Л.И. ТУРЧАК, П.В. ПЛОТНИКОВ

ОСНОВЫ ЧИСЛЕННЫХ МЕТОДОВ

ИЗДАНИЕ ВТОРОЕ, ПЕРЕРАБОТАННОЕ И ДОПОЛНЕННОЕ

Допущено Министерством образования
Российской Федерации в качестве учебного пособия
для студентов высших учебных заведений

МОСКВА
ФИЗМАТЛИТ
2003

УДК 519.6
ББК 22.19
Т89

Турчак Л. И., Плотников П. В. **Основы численных методов**: Учебное пособие. — 2-е изд., перераб. и доп. — М.: ФИЗМАТЛИТ, 2002. — 304 с. — ISBN 5-9221-0153-6.

Содержит основные сведения о численных методах, необходимые для первоначального знакомства с предметом. Излагаются основы численных методов для систем линейных и нелинейных уравнений, а также дифференциальных и интегральных уравнений. Имеется много задач, примеров и алгоритмов для облегчения понимания логической структуры рассматриваемых методов и их использования в расчетах на компьютерах.

Первое издание — 1987 г.

Для студентов вузов.

Табл. 21. Ил. 83. Библиогр. 66 назв.

ISBN 5-9221-0153-6

ОГЛАВЛЕНИЕ

ГЛАВА 3
ДИФФЕРЕНЦИРОВАНИЕ И ИНТЕГРИРОВАНИЕ

ГЛАВА 4
СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

ГЛАВА 5
НЕЛИНЕЙНЫЕ УРАВНЕНИЯ

ГЛАВА 6
МЕТОДЫ ОПТИМИЗАЦИИ

ГЛАВА 7
ОБЫКНОВЕННЫЕ ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ

ГЛАВА 8
УРАВНЕНИЯ С ЧАСТНЫМИ ПРОИЗВОДНЫМИ

ГЛАВА 9
ИНТЕГРАЛЬНЫЕ УРАВНЕНИЯ

ПРЕДИСЛОВИЕ

Внедрение компьютеров во все сферы человеческой деятельности требует от специалистов разного профиля овладения навыками использования вычислительной техники. Повышается уровень подготовки студентов вузов, которые уже с первых курсов приобщаются к использованию компьютеров и простейших численных методов, не говоря уже о том, что при выполнении курсовых и дипломных работ применение вычислительной техники стало нормой.

Применение компьютеров приобрело сейчас массовый характер. Они используются не только при научных и инженерных расчетах, но и для хранения и обработки информации, при решении ряда других задач и даже в быту. Тем не менее использование компьютера для проведения математических вычислений не потеряло своей актуальности. Причем оно распространилось не только на точные, технические и экономические науки, но и на такие традиционно нематематические специальности, как медицина, лингвистика, психология и др. Возникла многочисленная категория специалистов — пользователей компьютеров, использующих их в качестве вычислительного инструмента и поэтому нуждающихся в литературе по соответствующим дисциплинам.

Одной из основных дисциплин является вычислительная математика. Она изучает методы построения и исследования численных методов решения математических задач, которые моделируют различные процессы.

Численные методы разрабатывают и исследуют, как правило, высококвалифицированные специалисты – математики. Что касается подавляющей части студентов нематематических специальностей и инженерно – технических работников, то для них главным является понимание основных идей, методов, особенностей и областей их применения. Следует также иметь в виду, что указанная категория читателей не обладает достаточными математическими знаниями для подробного исследования численных методов. К тому же в этом нет особой необходимости специалисту – нематематику, использующему численные методы как готовый инструмент в своей практической работе.

В предлагаемом учебном пособии в сжатом виде приводятся основные необходимые сведения о численных методах решения различных прикладных задач. Изложение проводится на доступном для студентов втуза уровне. При необходимости напоминаются основные сведения из курса высшей

математики. Для многих рассматриваемых методов приводятся алгоритмы, а также примеры решения задач, способствующие лучшему пониманию материала. Книга написана с учетом особенностей применения численных методов при решении задач с использованием компьютеров.

Поскольку данное учебное пособие не ориентировано на студентов конкретной специальности, то приведенные в нем задачи носят общий характер. Большой выбор интересных задач содержится в книгах прикладного характера, включенных в список литературы. Они, в частности, могут быть использованы при выполнении курсовых и дипломных работ, а также в научно-исследовательской работе студентов. В список литературы включены также некоторые пособия по численным методам, которые авторы использовали в работе над данным пособием. Читатель может найти в них более подробные сведения по интересующим его разделам курса. Разумеется, это далеко не полный перечень литературы по численным методам и их приложениям.

При изложении материала сказался стиль чтения авторами курсов лекций для студентов нематематических специальностей втузов и слушателей факультета повышения квалификации. Книга будет полезна студентам и специалистам при первоначальном знакомстве с предметом. Она может служить кратким справочным пособием, которое студенты могут использовать при выполнении расчетных заданий.

Книга содержит девять глав. После каждой главы приведены упражнения для самостоятельного решения. Их выполнение читателем будет способствовать лучшему усвоению материала. Упражнения повышенной трудности отмечены звездочкой.

В главе 1 излагаются основные понятия, связанные с погрешностями вычислений. Рассматриваются источники погрешностей при расчетах на компьютерах.

Глава 2 посвящена различным способам аппроксимации (приближения) функций. При рассмотрении интерполирования дано понятие сплайнов, которые получили широкое распространение в вычислительной практике. Выписаны некоторые формулы, которые могут быть полезными при самостоятельной работе.

Вопросы численного дифференцирования и численного интегрирования изложены в главе 3. Здесь же приведены выражения для аппроксимаций производных, которые могут быть использованы при построении разностных схем для решения дифференциальных уравнений. Среди методов численного интегрирования упомянуто использование сплайнов. Приведено также понятие адаптивных алгоритмов, которые сейчас широко используются и при решении других задач.

Глава 4 содержит основные сведения по численному решению задач линейной алгебры. При первоначальном знакомстве можно опустить § 4, в котором излагаются некоторые задачи на собственные значения, поскольку эта тема носит специальный характер.

В главе 5 изложены основные методы решения нелинейных уравнений (алгебраических и трансцендентных) и их систем.

Глава 6, посвященная методам решения задач оптимизации, содержит также элементы линейного программирования.

Методы решения задач Коши и краевых задач для обыкновенных дифференциальных уравнений излагаются в главе 7.

В главе 8 излагаются численные методы решения уравнений с частными производными и приводятся некоторые элементы теории разностных схем.

Глава 9, посвященная интегральным уравнениям, носит ознакомительный характер, и при первом чтении может быть опущена. Вместе с тем следует отметить, что решение интегральных уравнений, в том числе и сингулярных, необходимо во многих областях науки (механике, физике и др.).

При подготовке ко второму изданию материал книги был переработан, дополнен изложением некоторых новых вопросов. Были исправлены замеченные опечатки, а также добавлена часть упражнений.

Авторы искренне признательны академику О. М. Белоцерковскому за ценные замечания по рукописи. Полезные предложения по улучшению содержания высказали З. С. Волк, Ю. Г. Евтушенко, И. К. Лифанов, В. Б. Миносцев, Г. П. Тиняков, Э. Г. Шифрин и другие товарищи, прочитавшие рукопись или отдельные ее части. Большую помощь в работе над книгой оказал В. В. Щенников. Всем им авторы выражают свою глубокую благодарность.

ВВЕДЕНИЕ

1. Этапы решения задачи на компьютере. Вычислительная техника нашла эффективное применение при проведении трудоемких расчетов в научных исследованиях. Действительно, современные компьютеры за одну секунду выполняют такой объем вычислений, на который человеку не хватит всей жизни.

При решении задачи на компьютере основная роль все-таки принадлежит человеку. Машина лишь выполняет его задания по разработанной программе. Роль человека и машины легко уяснить, если процесс решения задачи разбить на следующие этапы.

Постановка задачи. Этот этап заключается в содержательной (физической) постановке задачи и определении конечных целей решения.

Построение математической модели (математическая формулировка задачи). Модель должна правильно (адекватно) описывать основные законы физического процесса. Построение или выбор математической модели из существующих требует глубокого понимания проблемы и знания соответствующих разделов математики.

Разработка численного метода. Поскольку компьютер может выполнять лишь простейшие операции, он «не понимает» постановки задачи даже в математической формулировке. Для ее решения должен быть найден численный метод, позволяющий свести задачу к некоторому вычислительному алгоритму. Разработкой численных методов занимаются специалисты в области вычислительной математики. Специалисту – прикладнику для решения задачи, как правило, необходимо из имеющегося арсенала методов выбрать тот, который наиболее пригоден в данном конкретном случае.

Разработка алгоритма. Процесс решения задачи (вычислительный процесс) записывается в виде последовательности элементарных арифметических и логических операций, приводящей к конечному результату и называемой *алгоритмом* решения задачи. Алгоритм можно наглядно изобразить в виде блок-схемы, структурограммы и т. п. Опытный вычислитель зачастую может и не прибегать к такому наглядному представлению алгоритма, непосредственно переходя к следующему этапу.

Программирование. Алгоритм решения задачи записывается на понятном машине языке в виде точно определенной последовательности

операций — *программы* для компьютера. Составление программы (программирование) обычно производится с помощью некоторого промежуточного (алгоритмического) языка, а ее трансляция (перевод на машинный язык) осуществляется самой вычислительной системой.

О т л а д к а п р о г р а м м ы. Составленная программа содержит разного рода ошибки, неточности, описки. Отладка программы на машине включает контроль программы, диагностику (поиск и определение содержания) ошибок, их исправление. Программа испытывается на решении контрольных (тестовых) задач для получения уверенности в достоверности результатов.

П р о в е д е н и е р а с ч е т о в. На этом этапе готовятся исходные данные для расчетов и проводится счет по отлаженной программе. При этом для уменьшения ручного труда по обработке результатов желательно использовать удобные формы выдачи результатов, особенно их графическое представление (*визуализацию*).

А н а л и з р е з у л ь т а т о в. Результаты расчетов анализируются, оформляется научно-техническая документация.

Если при решении конкретной задачи возможно использование уже имеющихся прикладных программных средств, то некоторые из перечисленных этапов могут быть опущены. Так, для решения многих (хотя и достаточно узких) классов задач созданы программные продукты, существенно облегчающие труд вычислителя. Речь может идти, например, о том, что вычислитель сообщает программе только математическую модель (или даже постановку задачи) и исходные данные, а выбор метода, проведение расчетов, выдачу результатов программа берет на себя. Но даже в этом случае нельзя забывать о том, что полученное решение обычно является лишь приближенным, что каждая модель и каждый метод имеют свои области применимости. Следовательно, специалисту, использующему компьютер для решения прикладных задач, необходимо иметь представление об основах математического моделирования, численных методов, о возможностях компьютеров, уметь анализировать полученные результаты с точки зрения их достоверности.

Следует отметить еще один важный момент в процессе решения задачи с помощью компьютера. Это — *экономичность* выбранного способа решения задачи, численного метода, модели компьютера. В частности, если задача допускает простое аналитическое решение или измерение, то вряд ли целесообразно делать вычисления на машине. Иногда решение задачи производят с помощью большого вычислительного комплекса, хотя это можно было осуществить с использованием персонального компьютера.

Не умаляя значения физического эксперимента, нужно все-таки отметить неуклонно возрастающую долю вычислений на компьютере в общем объеме решения научно – технических задач. В связи с этим наряду с увеличением парка вычислительных машин и повышением их «интеллектуальных» возможностей возрастает интерес к математическому моделированию и разработке численных методов.

2. Математические модели. Основное требование, предъявляемое к математической модели, — *адекватность* рассматриваемому явлению, т. е. она должна достаточно точно (в рамках допустимых погрешностей) отражать характерные черты явления. Вместе с тем она должна обладать сравнительной простотой и доступностью исследования.

Приведем примеры некоторых математических моделей, оказавших огромное влияние на развитие различных отраслей науки и техники. При построении математических моделей получают некоторые математические соотношения (как правило, уравнения).

П р и м е р. Пусть в начальный момент времени $t = 0$ тело, находящееся на высоте h_0, начинает двигаться вертикально вниз с начальной скоростью v_0. Требуется найти закон движения тела, т. е. построить математическую модель, которая позволила бы математически описать данную задачу и определить параметры движения в любой момент времени.

Прежде чем строить указанную модель, нужно принять некоторые допущения, если они не заданы. В частности, предположим, что данное тело обладает средней плотностью, значительно превышающей плотность воздуха, а его форма близка к шару. В этом случае можно пренебречь сопротивлением воздуха и рассматривать свободное падение тела с учетом ускорения g. Соответствующие соотношения для высоты h и скорости v в любой момент времени t хорошо известны из школьного курса физики. Они имеют вид

$$h = h_0 - v_0 t - \frac{gt^2}{2}, \quad v = v_0 + gt. \tag{0.1}$$

Эти формулы являются искомой математической моделью свободного падения тела. Область применения данной модели ограничена случаями, в которых можно пренебречь сопротивлением воздуха. Во многих задачах о движении тел в атмосфере планеты модель (0.1) не может быть использована, поскольку при ее применении мы получили бы неверный результат. К таким задачам относятся движение капли, вход в атмосферу тел малой плотности, спуск на парашюте и др. Здесь необходимо построить более точную математическую модель, учитывающую сопротивление воздуха. Если обозначить через $F(t)$ силу сопротивления, действующую на тело массой m, то его движение можно описать с помощью уравнений

$$m\,\frac{dv}{dt} = mg - F(t), \quad \frac{dh}{dt} = -v. \tag{0.2}$$

К этой системе уравнений необходимо добавить начальные условия при $t = 0$:

$$v = v_0, \quad h = h_0. \tag{0.3}$$

Соотношения (0.2) и (0.3) являются математической моделью для задачи движения тела в атмосфере. Существуют и другие, более сложные

модели подобных задач (например, задача о движении планера). Заметим также, что модель (0.1) легко получается из (0.2), (0.3) при $F = 0$.

Известно большое число математических моделей различных процессов или явлений. Укажем некоторые из них, широко используемые в механике. Модель абсолютно твердого тела позволила получить уравнения движения тел в динамике полета. Модель идеального газа привела к системе уравнений Эйлера, описывающей невязкие потоки газов. В гидродинамике широко известна модель на основе уравнений Навье – Стокса, в кинетической теории газов — уравнения Больцмана. В механике деформируемого твердого тела известны математические модели, описывающие различные среды (упругую, упруго – пластичную и др.).

Имеются математические модели и для описания задач экономики, социологии, медицины, лингвистики и др.

Адекватность и сравнительная простота модели не исчерпывают предъявляемых к ней требований. Обратим еще внимание на необходимость правильной оценки области применимости математической модели. Например, модель свободно падающего тела, в которой пренебрегают сопротивлением воздуха, весьма эффективна для твердых тел с большой средней плотностью и формой поверхности, близкой к сферической. Вместе с тем в ряде других случаев (движения капельки жидкости, парашютного устройства и др.) для решения задачи уже недостаточно известных из курса физики простейших формул. Здесь необходимы более сложные математические модели, учитывающие сопротивление воздуха и другие факторы.

Отметим, что успех решения задачи в значительной степени определяется выбором математической модели: здесь в первую очередь нужны глубокие знания в той области, к которой принадлежит поставленная задача. Кроме того, необходимы знания соответствующих разделов математики и возможностей компьютеров.

3. Численные методы. С помощью математического моделирования решение научно-технической задачи сводится к решению математической задачи, являющейся ее моделью. Для решения математических задач используются следующие основные группы методов: аналитические, графические и численные.

При использовании *аналитических методов* решение задачи удается выразить с помощью формул. В частности, если математическая задача состоит в решении простейших алгебраических или трансцендентных уравнений, дифференциальных уравнений и т. п., то использование известных из курса математики приемов сразу приводит к цели. К сожалению, на практике это бывает достаточно редко.

Графические методы позволяют в ряде случаев оценить порядок искомой величины. Основная идея этих методов состоит в том, что решение находится путем геометрических построений. Например, для нахождения корней уравнения $f(x) = 0$ строится график функции $y = f(x)$, точки пересечения которого с осью абсцисс и будут искомыми корнями.

Графические методы могут применяться для получения начального приближения к решению, которое затем уточняется с помощью численных методов.

Основным инструментом для решения сложных математических задач в настоящее время являются *численные методы*, позволяющие свести решение задачи к выполнению конечного числа арифметических действий над числами; при этом результаты получаются в виде числовых значений.

Подчеркнем важные отличия численных методов от аналитических. Во-первых, численные методы позволяют получить лишь приближенное решение задачи. Во-вторых, они обычно позволяют получить лишь решение задачи с конкретными значениями параметров и исходных данных.

Поясним второе отличие на примере. По формулам (0.1) (по аналитическому решению) можно проанализировать, как изменится закон движения при изменении параметра g и начальных значений v_0, h_0. Если в модели (0.2), (0.3) выражение для $F(t)$ имеет простой вид (например, $F = \mathrm{const}$), то можно получить аналитическое решение, аналогичное (0.1). Это решение тоже легко исследовать на предмет зависимости от изменения параметров и начальных условий. Если же выражение для $F(t)$ достаточно сложно, то задачу (0.2), (0.3) можно решить только численно. При этом вместо общей формулы решения в результате расчета будут получены значения v и h для некоторого набора моментов времени t при конкретных значениях g, m, v_0, h_0. Для получения решения при других значениях параметров и (или) других начальных условиях необходимо провести новый расчет. Для анализа зависимости решения от параметров и начальных условий необходима большая серия расчетов.

Несмотря на эти недостатки, численные методы незаменимы в сложных задачах, которые не допускают аналитического решения.

Многие численные методы разработаны давно, однако при вычислениях вручную они могли использоваться лишь для решения не слишком трудоемких задач. С появлением компьютеров начался период бурного развития численных методов и их внедрения в практику. Только вычислительной машине под силу выполнить за короткое время объем вычислений в миллиарды, триллионы и более операций, необходимых для решения многих современных задач.

Численный метод наряду с возможностью получения результата за приемлемое время должен обладать и еще одним важным качеством — не вносить в вычислительный процесс значительных погрешностей.

ГЛАВА 1

ТОЧНОСТЬ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

§ 1. Приближенные числа

1. Числа с плавающей точкой. Числа могут быть представлены в памяти компьютеров различными способами. Современные компьютеры (процессоры), как правило, позволяют обрабатывать *целые* числа, а также дробные числа в форме с *плавающей точкой* [1].

Как известно, множество целых чисел бесконечно. Однако процессор из-за ограниченности его *разрядной сетки* может оперировать лишь с некоторым конечным подмножеством этого множества. В современных компьютерах для хранения целого числа обычно отводится 4 *байта* памяти [2], что позволяет представлять целые числа, находящиеся примерно в диапазоне от $-2 \cdot 10^9$ до $2 \cdot 10^9$.

При решении научно-технических задач в основном используются действительные (вещественные) числа. В компьютерах они представляются в форме с плавающей точкой. Десятичное число D в этой форме записи имеет вид $D = \pm m \cdot 10^n$, где m и n — соответственно *мантисса* числа и его *порядок*. Например, число -273.9 можно записать в виде: $-2739 \cdot 10^{-1}$, $-2.739 \cdot 10^2$, $-0.2739 \cdot 10^3$. Последняя запись — нормализованная форма числа с плавающей точкой. Таким образом, если представить мантиссу числа в виде $m = 0.\, d_1 d_2 \ldots d_k$, то при $d_1 \neq 0$ получим *нормализованную форму* числа с плавающей точкой. В дальнейшем, говоря о числах с плавающей точкой, будем иметь в виду именно эту форму. Обычная же запись числа в виде -273.9 называется формой записи с *фиксированной точкой*. В настоящее время такое представление используется в компьютерах, как правило, только на этапе ввода и вывода чисел.

Все сказанное выше о числах с плавающей точкой распространяется и на числа, записанные в других системах счисления. Число A в системе счисления с основанием α можно представить в виде $A = \pm 0.\, a_1 a_2 \ldots a_k \cdot \alpha^n$, где $a_1, a_2, \ldots, a_k$ — целые числа из диапазона $0, \ldots, \alpha - 1$. Из этой записи следует, что подмножество действительных чисел, с которым оперирует

---

[1] В англоязычных странах целую и дробную части при десятичной записи числа разделяют точкой, а не запятой. Часто аналогично поступают и в специализированной литературе на русском языке.

[2] Здесь и далее при упоминании характеристик компьютеров имеются в виду прежде всего широко распространенные персональные компьютеры на базе процессоров фирмы Intel.

конкретный компьютер, не является бесконечным: оно конечно и определяется разрядностью k, а также границами порядка n_1, n_2 ($n_1 \leqslant n \leqslant n_2$). Можно показать, что это подмножество содержит

$$N = 2(\alpha - 1)(n_2 - n_1 + 1)\alpha^{k-1} + 1 \tag{1.1}$$

чисел, наименьшим и наибольшим по модулю являются соответственно числа

$$M_0 = (\alpha - 1)\alpha^{n_1 - 1} \quad \text{и} \quad M_\infty = (1 - \alpha^{-k})\alpha^{n_2}, \tag{1.2}$$

называемые *машинным нулем* и *машинной бесконечностью*.

Границы порядка n_1, n_2 определяют ограниченность действительных чисел по величине, а разрядность k — дискретность распределения их на отрезке числовой оси. Например, в случае десятичных чисел при четырехразрядном представлении все значения, находящиеся в промежутке $(0.28505, 0.28515)$, представляются числом 0.2851 (при выполнении округления). Если к этому числу 0.2851 прибавить число, меньшее по модулю половины единицы последнего разряда (т. е. меньшее по модулю 0.00005), в результате получится то же самое число 0.2851.

В настоящее время большинство производителей процессоров в основном придерживаются стандарта IEEE 754 [1)] для арифметических операций над двоичными числами с плавающей точкой. Данный стандарт предусматривает наличие, в частности, двух двоичных ($\alpha = 2$) форматов: с одинарной точностью и с двойной точностью. Приведем для этих форматов размер отводимой памяти, значения k, n_1, n_2 и приближенные значения M_0 и M_∞. Заметим, что стандарт IEEE 754 предусматривает обработку чисел, меньших по модулю M_0, но не меньших M_0^*, правда, с меньшей разрядностью k.

Точность	Байты	k	n_1	n_2	M_0	M_0^*	M_∞
Одинарная	4	24	-125	128	$1.2 \cdot 10^{-38}$	$1.4 \cdot 10^{-45}$	$3.4 \cdot 10^{38}$
Двойная	8	53	-1021	1024	$2.2 \cdot 10^{-308}$	$4.9 \cdot 10^{-324}$	$1.8 \cdot 10^{308}$

Поскольку для человека более удобной является десятичная система счисления, возникает вопрос о том, скольким десятичным разрядам соответствует указанная двоичная разрядность k. Можно считать, что k соответствует 6 – 9 десятичным разрядам при одинарной и 15–17 разрядам при двойной точности.

В современных языках программирования предусмотрены типы данных для представления вещественных чисел с одинарной и двойной точностью. Например, в языке Си это типы float и double, в языке Паскаль — single и double, в языке Фортран — real и double precision. Обычно эти представления соответствуют стандарту IEEE 754.

[1)] IEEE — институт инженеров по электротехнике и электронике (США).

Таким образом, компьютер оперирует с приближенными значениями действительных чисел. Мерой точности приближенных чисел является погрешность.

2. Понятие погрешности. Различают два вида погрешностей — абсолютную и относительную. *Абсолютная погрешность* некоторого числа равна разности между его истинным значением и приближенным значением, полученным в результате вычисления или измерения. *Относительная погрешность* — это отношение абсолютной погрешности к приближенному значению числа.

Таким образом, если a — приближенное значение числа x, то выражения для абсолютной и относительной погрешностей запишутся соответственно в виде

$$\Delta x = x - a, \quad \delta x = \Delta x / a.$$

К сожалению, истинное значение величины x обычно неизвестно. Поэтому приведенные выражения для погрешностей практически не могут быть использованы. Имеется лишь приближенное значение a и нужно найти его *предельную погрешность* Δa, являющуюся верхней оценкой модуля абсолютной погрешности, т. е. $|\Delta x| \leqslant \Delta a$. В дальнейшем значение Δa принимается в качестве абсолютной погрешности приближенного числа a. В этом случае истинное значение x находится в интервале $(a - \Delta a, a + \Delta a)$.

Для приближенного числа, полученного в результате округления, абсолютная погрешность Δa принимается равной половине единицы последнего разряда числа. Например, значение $a = 0.734$ могло быть получено округлением чисел 0.73441, 0.73353 и др. При этом $|\Delta x| \leqslant 0.0005$, и полагаем $\Delta a = 0.0005$. Если при вычислениях на компьютере округление не производится, а цифры, выходящие за разрядную сетку машины, отбрасываются, то максимально возможная погрешность результата выполнения операции в два раза больше по сравнению со случаем округления.

Приведем примеры оценки абсолютной погрешности при некоторых значениях приближенной величины a:

a	51.7	-0.0031	16	16.00
Δa	0.05	0.00005	0.5	0.005

Предельное значение относительной погрешности — отношение предельной абсолютной погрешности к абсолютной величине приближенного числа:

$$\delta a = \Delta a / |a|.$$

Например, $\delta(-2.3) = 0.05/2.3 \approx 0.022$ $(2.2\,\%)$. Заметим, что погрешность округляется всегда в сторону увеличения. В данном случае $\delta(-2.3) \approx 0.03$.

Приведенные оценки погрешностей приближенных чисел справедливы, если в записи этих чисел все значащие цифры верные. Напомним, что *значащими цифрами* считаются все цифры данного числа, начиная с первой ненулевой цифры. Например, в числе 0.037 две значащие цифры: 3 и 7,

а в числе 14.80 все четыре цифры значащие. Кроме того, при изменении формы записи числа (например, при записи в форме с плавающей точкой) число значащих цифр не должно меняться, т. е. нужно соблюдать равносильность преобразований. Например, записи $7500 = 0.7500 \cdot 10^4$ и $0.110 \cdot 10^2 = 11.0$ равносильные, а записи $7500 = 0.75 \cdot 10^4$ и $0.110 \cdot \cdot 10^2 = 11$ неравносильные.

3. Действия над приближенными числами. Сформулируем правила оценки предельных погрешностей при выполнении операций над приближенными числами.

При сложении или вычитании чисел их абсолютные погрешности складываются. При умножении или делении чисел друг на друга их относительные погрешности складываются. При возведении в степень приближенного числа его относительная погрешность умножается на показатель степени.

Для случая двух приближенных чисел a и b эти правила можно записать в виде формул

$$\begin{gathered} \Delta(a \pm b) = \Delta a + \Delta b, \quad \delta(a \cdot b) = \delta a + \delta b, \\ \delta(a/b) = \delta a + \delta b, \quad \delta(a^k) = k\delta a. \end{gathered} \tag{1.3}$$

Относительная погрешность суммы положительных слагаемых заключена между наибольшим и наименьшим значениями относительных погрешностей этих слагаемых. Действительно, пусть $a > 0$, $b > 0$, $m = \min(\delta a, \delta b)$, $M = \max(\delta a, \delta b)$. Тогда

$$\delta(a+b) = \frac{\Delta(a+b)}{a+b} = \frac{\Delta a + \Delta b}{a+b} = \frac{a\delta a + b\delta b}{a+b} \leqslant \frac{aM + bM}{a+b} = M.$$

Аналогично, $\delta(a+b) \geqslant m$. На практике для оценки погрешности принимается наибольшее значение M.

Пример 1. Найти относительную погрешность функции

$$y = \sqrt{\frac{a+b}{x^3(1-x)}}.$$

Используя формулы (1.3), получаем

$$\delta y = \frac{1}{2}\left[\delta(a+b) + 3\delta x + \delta(1-x)\right] = \frac{1}{2}\left[\frac{\Delta a + \Delta b}{|a+b|} + 3\,\frac{\Delta x}{|x|} + \frac{\Delta x}{|1-x|}\right].$$

Полученная оценка относительной погрешности содержит в знаменателе выражение $|1-x|$. Ясно, что при $x \approx 1$ можно получить очень большую погрешность. В связи с этим рассмотрим подробнее случай вычитания близких чисел.

Запишем выражение для относительной погрешности разности двух чисел в виде

$$\delta(a-b)=\frac{\Delta(a-b)}{|a-b|}=\frac{\Delta a+\Delta b}{|a-b|}.$$

При $a\approx b$ эта погрешность может быть сколь угодно большой.

Пример 2. Пусть $a=2520$, $b=2518$. В этом случае имеем абсолютные погрешности исходных данных $\Delta a=\Delta b=0.5$ и относительные погрешности $\delta a\approx\delta b=0.5/2518\approx 0.0002$ $(0.02\,\%)$. Относительная погрешность разности равна

$$\delta(a-b)=\frac{0.5+0.5}{2}=0.5\ (50\,\%).$$

Следовательно, при малых погрешностях в исходных данных мы получили весьма неточный результат. Нетрудно подсчитать, что даже при случайных изменениях a и b на единицу в последних разрядах их разность может принимать значения 0, 1, 2, 3, 4. Поэтому при организации вычислительных алгоритмов следует избегать вычитания близких чисел; при возможности алгоритм нужно видоизменить во избежание потери точности на некотором этапе вычислений.

Из рассмотренных правил следует, что при сложении или вычитании приближенных чисел желательно, чтобы эти числа обладали одинаковыми абсолютными погрешностями, т. е. одинаковым числом разрядов после десятичной точки. Например, $38.723+4.9=43.6$; $425.4-0.047=425.4$. Учет отброшенных разрядов не повысит точность результатов. При умножении и делении приближенных чисел количество значащих цифр выравнивается по наименьшему из них.

Наряду с приведенными выше оценками погрешностей при выполнении некоторых операций над приближенными числами можно записать аналогичные оценки и для вычисления функций, аргументами которых являются приближенные числа. Однако более полным оказывается общее правило, основанное на вычислении приращения (погрешности) функции при заданных приращениях (погрешностях) аргументов.

Рассмотрим функцию одной переменной $y=f(x)$. Пусть a — приближенное значение аргумента x, Δa — его абсолютная погрешность. Абсолютную погрешность функции можно считать ее приращением, которое она испытывает при изменении аргумента на Δa. Это приращение можно заменить дифференциалом: $\Delta y\approx dy$. Тогда для оценки абсолютной погрешности получим выражение $\Delta y=|f'(a)|\Delta a$.

Аналогичное выражение можно записать для функции нескольких аргументов. Например, оценка абсолютной погрешности функции $u=f(x,y,z)$, приближенные значения аргументов которой соответственно a, b, c, имеет вид

$$\Delta u=|f'_x(x,y,z)|\Delta a+|f'_y(x,y,z)|\Delta b+|f'_z(x,y,z)|\Delta c. \qquad (1.4)$$

Здесь Δa, Δb, Δc — абсолютные погрешности аргументов. Относительная погрешность находится по формуле

$$\delta u = \frac{\Delta u}{|f(a,b,c)|}.$$

Полученные соотношения можно использовать для вывода оценки погрешности произвольной функции (таким способом легко получить выражения (1.3)). Например, при $c = a - b$ по формуле (1.4) получаем $\Delta c = |c'_a|\Delta a + |c'_b|\Delta b = \Delta a + \Delta b$.

§ 2. Погрешности вычислений

1. Источники погрешностей. На некоторых этапах решения задачи на компьютере могут возникать погрешности, искажающие результаты вычислений. Оценка степени достоверности получаемых результатов является важнейшим вопросом при организации вычислительных работ. Это особенно важно при отсутствии опытных или других данных для сравнения, которое могло бы в некоторой степени показать надежность используемого численного метода и достоверность получаемых результатов.

Рассмотрим источники погрешностей на отдельных этапах решения задачи.

Математическая модель, принятая для описания данного процесса или явления, может внести существенные погрешности, если в ней не учтены какие-либо важные черты рассматриваемой задачи. В частности, математическая модель может прекрасно работать в одних условиях и быть совершенно неприемлемой в других; поэтому важно правильно учитывать область ее применимости.

Исходные данные задачи часто являются основным источником погрешностей. Вместе с погрешностями, вносимыми математической моделью, их называют *неустранимыми погрешностями*, поскольку они не могут быть уменьшены вычислителем ни до начала решения задачи, ни в процессе ее решения. Проведенный ранее анализ оценки погрешностей при выполнении арифметических операций показывает, что следует стремиться к тому, чтобы все исходные данные были примерно одинаковой точности. Сильное уточнение одних исходных данных при наличии больших погрешностей в других, как правило, не приводит к повышению точности результатов.

Численный метод также является источником погрешностей. Это связано, например, с заменой интеграла суммой, с усечением рядов при вычислениях значений функций, с интерполированием табличных данных и т. п. Как правило, *погрешность численного метода* регулируема, т. е. теоретически она может быть уменьшена до любого значения путем изменения некоторого параметра (например, шага интегрирования, числа членов усеченного ряда и т. п.). Погрешность метода обычно стараются

довести до величины, в несколько раз меньшей неустранимой погрешности. Дальнейшее снижение погрешности не приведет к повышению точности результатов, а лишь увеличит стоимость расчетов из-за необоснованного увеличения объема вычислений. Подробнее погрешности методов будем рассматривать при анализе конкретных численных методов.

При вычислениях с помощью компьютера неизбежны *погрешности округлений*, связанные с ограниченностью разрядной сетки машины. При обычном округлении (которое, как правило, и реализуется в компьютерах) максимальная относительная погрешность есть

$$\delta_{\max} = 0.5\alpha^{1-k}, \tag{1.5}$$

где α — основание системы счисления, k — количество разрядов мантиссы числа. При простом отбрасывании лишних разрядов эта погрешность увеличивается вдвое.

Вычислим по формуле (1.5) максимальную погрешность округления $\delta_{\max}$ для чисел, представленных в форматах с одинарной и двойной точностью стандарта IEEE 754. Имеем: $\alpha = 2$ в обоих случаях, для одинарной точности $k = 24$ и $\delta_{\max} \approx 6 \cdot 10^{-8}$, для двойной точности $k = 53$ и $\delta_{\max} \approx 10^{-16}$.

Несмотря на то, что при решении больших задач выполняются миллиарды и триллионы операций, это вовсе не означает механического умножения погрешности при одном округлении на число операций, так как при отдельных действиях погрешности могут компенсировать друг друга (например, при сложении чисел разных знаков). Вместе с тем иногда погрешности округлений в сочетании с плохо организованным алгоритмом могут сильно исказить результаты. В дальнейшем мы рассмотрим такие случаи.

Перевод чисел из одной системы счисления в другую также может быть источником погрешности из-за того, что основание одной системы счисления не является степенью основания другой (например, 10 и 2). Это может привести к тому, что в новой системе счисления число становится иррациональным.

Например, число 0.1 при переводе в двоичную систему счисления примет вид $0.000\,1100\,1100\ldots$ Может оказаться, что с шагом 0.1 нужно при вычислениях пройти отрезок $[0, 1]$ от $x = 1$ до $x = 0$; десять шагов не дадут точного значения $x = 0$.

2. Уменьшение погрешностей. При рассмотрении погрешностей результатов арифметических операций отмечалось, что вычитание близких чисел приводит к увеличению относительной погрешности; поэтому в алгоритмах следует избегать подобных ситуаций. Рассмотрим также некоторые другие случаи, когда можно избежать потери точности правильной организацией вычислений.

Пусть требуется найти сумму пяти четырехразрядных чисел: $S = 0.2764 + 0.3944 + 1.475 + 26.46 + 1364$. Складывая все эти числа, а затем округляя полученный результат до четырех значащих цифр, получаем $S = 1393$. Однако при вычислении на компьютере округление происходит

после каждого сложения. Предполагая условно сетку четырехразрядной, проследим за вычислением на компьютере суммы чисел от наименьшего к наибольшему, т. е. в порядке их записи: $0.2764 + 0.3944 = 0.6708$, $0.6708 + 1.475 = 2.156$, $2.156 + 26.46 = 28.62$, $28.62 + 1364 = 1393$; получили $S_1 = 1393$, т. е. верный результат. Изменим теперь порядок вычислений и начнем складывать числа последовательно от последнего к первому: $1364+26.46 = 1390$, $1390+1.475 = 1391$, $1391+0.3944 = 1391$, $1391+0.2764 = 1391$; здесь окончательный результат $S_2 = 1391$, он менее точный.

Анализ процесса вычислений показывает, что потеря точности здесь происходит из-за того, что прибавления к большому числу малых чисел не происходит, поскольку они выходят за рамки разрядной сетки ($a+b = a$ при $a \gg b$) . Этих малых чисел может быть очень много, но на результат они все равно не повлияют, поскольку прибавляются по одному, что и имело место при вычислении S_2. Здесь необходимо придерживаться правила, в соответствии с которым сложение чисел нужно проводить по мере их возрастания. В машинной арифметике из-за погрешности округления существен порядок выполнения операций, и известные из алгебры законы коммутативности (и дистрибутивности) здесь не всегда выполняются.

При решении задачи на компьютере нужно использовать подобного рода маленькие хитрости для улучшения алгоритма и снижения погрешностей результатов. Например, при вычислении на компьютере значения $(a + x)^2$ величина x может оказаться такой, что результатом сложения $a + x$ получится a (при $x \ll a$); в этом случае может помочь замена $(a + x)^2 = a^2 + 2ax + x^2 = a(a + 2x) + x^2$. Действительно, теперь к a прибавляется не x, а $2x$. Если же при $x \ll a$ вычисляется величина $(a + x)^2 - a^2$, то целесообразно привести ее к виду $2ax + x^2$, избежав тем самым вычитания близких величин.

Рассмотрим еще один важный пример — использование рядов для вычисления значений функций. Запишем, например, разложение функции $\sin x$ по степеням аргумента:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

По признаку Лейбница остаток сходящегося знакочередующегося ряда, т. е. погрешность суммы конечного числа членов, не превышает значения первого из отброшенных членов (по абсолютной величине).

Вычислим значение функции $\sin x$ при $x = 0.5236$ (30°). Члены ряда, меньшие 10^{-4}, не будем учитывать. Вычисления проведем с четырьмя верными знаками. Получим

$$\sin 0.5236 = 0.5236 - 0.2392 \cdot 10^{-1} + 0.3279 \cdot 10^{-3} = 0.500.$$

Это отличный результат в рамках принятой точности. Зная из курса высшей математики, что это разложение синуса справедливо при любом значении аргумента $(-\infty < x < +\infty)$, используем его для вычисления функции

при $x = 6.807$ (390°). Опуская вычисления, получаем $\sin 6.807 \approx 0.5167$. Относительная погрешность составляет здесь около $3\,\%$ (вместо ожидаемого значения $0.01\,\%$ по признаку Лейбница). Это объясняется погрешностями округлений и способом суммирования ряда (слева направо, без учета величины членов).

Не всегда помогает и повышенная точность вычислений. В частности, для данного ряда при $x = 25.6563\ldots$ $(1470° = 4 \cdot 360° + 30°)$ даже при учете членов ряда до 10^{-8} и вычислениях с восемью значащими цифрами в результате аналогичных вычислений (суммирование слева направо) получается результат, не имеющий смысла: $\sin x \approx 129$.

В программах, использующих степенные ряды для вычисления значений функций, могут быть приняты различные меры по предотвращению подобной потери точности. Так, влияние погрешностей округления существенно уменьшается, если $|x| < 1$. Действительно, при вычислении x^k допускается абсолютная погрешность

$$\Delta(x^k) = x^k \delta(x^k) = x^k k \delta x$$

(см. (1.3)), которая при невыполнении неравенства $|x| < 1$ может стать неприемлемо большой. Для тригонометрических функций можно использовать формулы приведения, благодаря чему аргумент будет находиться на отрезке $[0,\ 1]$. При вычислении экспоненты аргумент x можно разбить на сумму целой и дробной частей ($e^x = e^{n+a} = e^n \cdot e^a$, $0 < a < 1$) и использовать разложение в ряд только для e^a, а e^n вычислять умножением. Таким образом, при организации вычислений можно своевременно распознать «подводные камни», дающие потерю точности, и попытаться затем исправить положение.

3. О решении квадратного уравнения. Мы убедились в том, что при численном решении задач на компьютере вычислителя ожидают всякие «ловушки», которые могут привести к заметной потере точности результатов или даже к прекращению счета. Хорошей иллюстрацией к этому является анализ алгоритма решения такой простой задачи, как решение квадратного уравнения $ax^2 + bx + c = 0$. Его корни определяются соотношениями

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_2 = \frac{-b + \sqrt{D}}{2a}, \quad D = b^2 - 4ac. \tag{1.6}$$

Из анализа этих формул видно, что здесь имеется ряд особенностей вычислительного характера, которые необходимо иметь в виду при составлении алгоритма.

Рассмотрим простейший случай $a = 0$. Здесь уравнение становится линейным, и его единственный корень есть $x = -c/b$, если $b \neq 0$. При $a = b = 0$ и $c \neq 0$ уравнение не имеет решения, а в случае $a = b = c = 0$ его решением будет любое число. Заметим, что в машинной арифметике редко получаются точно нулевые значения. Поэтому коэффициенты можно сравнивать не с нулем, а с некоторой малой величиной ε.

Это в свою очередь порождает ряд ситуаций, зависящих от соотношения между коэффициентами.

Далее необходимо предусмотреть разветвление алгоритма в зависимости от знака дискриминанта D: $D > 0$ — корни действительные (см. (1.6)); $D = 0$ — корни равные: $x_1 = x_2 = -b/(2a)$; $D < 0$ — корни комплексные: $x_{1,2} = R \pm iI$, где $R = -b/(2a)$, $I = \sqrt{-D}/(2a)$.

Менее очевидным вопросом является возможность появления погрешностей в зависимости от соотношения между коэффициентами уравнения. Рассмотрим один из важнейших случаев, когда коэффициент b значительно превышает по абсолютной величине остальные. При этом $b^2 \gg 4ac$ и возникает опасность вычитания близких чисел в числителе одного из выражений (1.6) из-за того, что $\sqrt{D} \approx |b|$.

Положение можно исправить разными способами. Например, при $b > 0$ формулу для x_2 можно преобразовать следующим образом:

$$x_2 = \frac{\sqrt{D} - b}{2a} \frac{\sqrt{D} + b}{\sqrt{D} + b} = -\frac{2c}{\sqrt{D} + b}.$$

При $b < 0$ аналогичным способом можно записать формулу для x_1.

Более универсальным способом является использование значения $\operatorname{sign} b$ («знак величины b»):

$$\operatorname{sign} b = \begin{cases} 1, & b \geqslant 0, \\ -1, & b < 0. \end{cases} \tag{1.7}$$

Тогда один из корней может быть вычислен по формуле

$$x_1 = -\frac{b + \operatorname{sign} b \cdot \sqrt{D}}{2a}. \tag{1.8}$$

Выражение для вычисления значения второго корня можно получить с помощью теоремы Виета. Из соотношения $x_1 x_2 = c/a$ следует, что

$$x_2 = \frac{c}{a x_1}. \tag{1.9}$$

На рис. 1.1 и 1.2 представлены структурограмма (см. приложение А) и (для сравнения) блок-схема одного из вариантов алгоритма решения квадратного уравнения с учетом рассмотренных здесь особенностей. При $D > 0$ значения корней вычисляются по формулам (1.8), (1.9). Заметим, что в приведенном на структурограмме алгоритме предусмотрены еще не все случаи возможных вычислительных затруднений, которые могут встретиться при решении квадратных уравнений.

Можно привести некоторые примеры, когда реализация этого алгоритма на компьютере невозможна. Будем предполагать, что вычисления проводятся с двойной точностью.

Пример 1. $a = 10^{-200}$, $b = -3 \cdot 10^{-200}$, $c = 2 \cdot 10^{-200}$. При вычислении произведений b^2 и $4ac$ получается машинный нуль, т. е. $D = 0$;

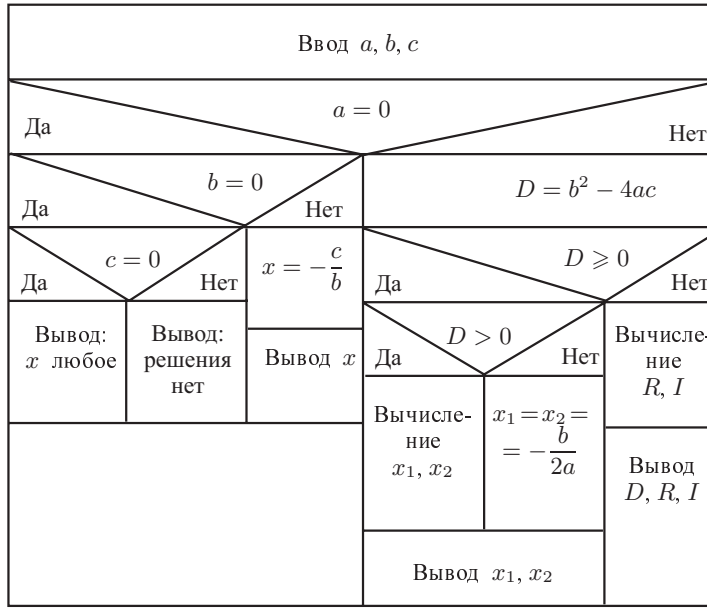


Рис. 1.1. Структурограмма алгоритма решения квадратного уравнения

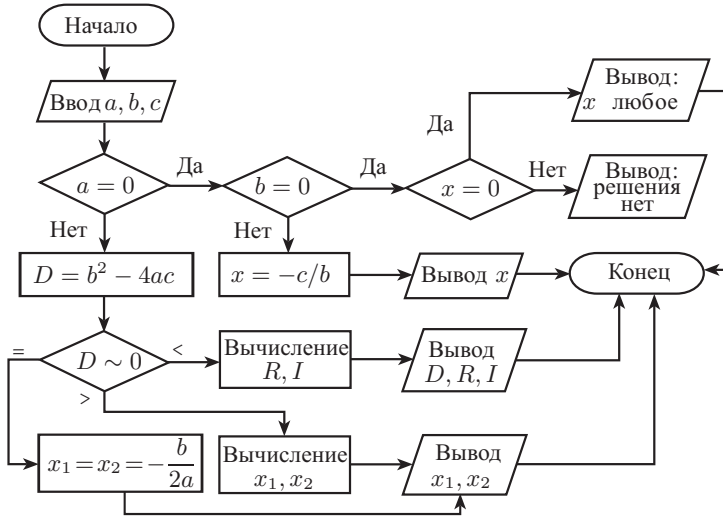


Рис. 1.2. Блок–схема алгоритма решения квадратного уравнения

решение пойдет по ветви равных корней: $x_1 = x_2 = 1.5$. Точные значения корней, как нетрудно видеть, равны $x_1 = 1,\ x_2 = 2$.

Пример 2. $a = 10^{200}$, $b = -3 \cdot 10^{200}$, $c = 2 \cdot 10^{-200}$. Этот вариант аналогичен предыдущему случаю с той лишь разницей, что вместо получения машинного нуля произойдет переполнение и прерывание счета.

Пример 3. $a = 10^{-200}$, $b = 10^{200}$, $c = -10^{200}$. Это трудный для реализации на компьютере случай. В практических расчетах встречаются уравнения с малым коэффициентом при x^2. В этом случае $b^2 \gg 4ac$, но при вычислении b^2 произойдет переполнение. Простейшим выходом из этого положения может быть сведение к случаю $a = 0$ с обязательной проверкой других коэффициентов.

Таким образом, анализ даже такой задачи, как решение квадратного уравнения, показывает, что использование численного алгоритма может быть сопряжено с некоторыми трудностями.

§ 3. Устойчивость. Корректность. Сходимость

1. Устойчивость. Рассмотрим погрешности исходных данных. Поскольку это так называемые неустранимые погрешности и вычислитель не может с ними бороться, то нужно хотя бы иметь представление об их влиянии на точность окончательных результатов. Конечно, мы вправе надеяться на то, что погрешность результатов имеет порядок погрешности исходных данных. Всегда ли это так? К сожалению, нет. Некоторые задачи весьма чувствительны к неточностям в исходных данных. Эта чувствительность характеризуется так называемой устойчивостью.

Пусть в результате решения задачи по исходному значению величины x находится значение искомой величины y. Если исходная величина имеет абсолютную погрешность Δx, то решение имеет погрешность Δy. Задача называется *устойчивой* по исходному параметру x, если решение y непрерывно от него зависит, т. е. малое приращение исходной величины Δx приводит к малому приращению искомой величины Δy. Другими словами, малые погрешности в исходной величине приводят к малым погрешностям в решении.

Отсутствие устойчивости означает, что даже незначительные погрешности в исходных данных приводят к большим погрешностям в решении или даже к неверному результату. О неустойчивых задачах также говорят, что они *чувствительны* к погрешностям исходных данных.

Приведем пример неустойчивой задачи. Рассмотрим квадратное уравнение с параметром a

$$x^2 - 2x + \operatorname{sign} a = 0.$$

Функция sign определена в (1.7). Решение этого уравнения в зависимости от значения a таково: $x_1 = x_2 = 1$ при $a \geqslant 0$; $x_{1,2} = 1 \pm \sqrt{2}$ при $a < 0$. Очевидно, что при $a = 0$ сколь угодно малая отрицательная погрешность в задании a приведет к конечной, а не сколь угодно малой погрешности в решении уравнения.

Иногда бывает, что теоретически задача устойчива, но тем не менее чувствительна к погрешностям исходным данных. Приращения исходной величины, гарантирующие малость приращения искомой величины, оказываются в этом случае настолько малыми, что реальные малые приращения, с которыми имеет дело вычислитель, приводят к большим погрешностям в решении.

Интересной иллюстрацией такой задачи является так называемый *пример Уилкинсона*. Рассматривается многочлен

$$P(x) = (x-1)(x-2)\dots(x-20) = x^{20} - 210x^{19} + \dots$$

Очевидно, что корнями этого многочлена являются $x_1 = 1$, $x_2 = 2$, $\dots$ $\dots$, $x_{20} = 20$. Предположим, что один из коэффициентов многочлена вычислен с некоторой малой погрешностью. Например, коэффициент -210 при x^{19} увеличим на 10^{-7}. В результате вычислений с двойной точностью получим существенно другие значения корней. Приведем для наглядности эти значения, округленные до трех значащих цифр:

$$\begin{aligned}
x_1 &= 1.00, & x_9 &= 8.93,\\
x_2 &= 2.00, & x_{10,11} &= 10.1 \pm 0.601i,\\
x_3 &= 3.00, & x_{12,13} &= 11.8 \pm 1.60i,\\
x_4 &= 4.00, & x_{14,15} &= 14.0 \pm 2.45i,\\
x_5 &= 5.00, & x_{16,17} &= 16.7 \pm 2.73i,\\
x_6 &= 6.00, & x_{18,19} &= 19.5 \pm 1.87i,\\
x_7 &= 7.00, & x_{20} &= 20.8.\\
x_8 &= 8.01,
\end{aligned}$$

Таким образом, изменение коэффициента при x^{19} с -210 до $-210 + 10^{-7}$ (а это, несомненно, малое изменение в обычной вычислительной практике) привело к тому, что половина корней стали комплексными. Причина такого явления — чувствительность задачи к погрешностям исходным данных; вычисления выполнялись достаточно точно, и погрешности округления не могли привести к таким последствиям. Заметим, что если коэффициент -210 изменить на значительно меньшее число, чем 10^{-7}, то изменение значений корней станет малым. Например, при увеличении коэффициента -210 на 10^{-11} значения корней, округленные до трех знаков, совпадут со значениями корней исходного многочлена. Примерно такого результата и следовало ожидать, поскольку рассматриваемая нами задача устойчива.

2. Корректность. Задача называется *поставленной корректно*, если для любых значений исходных данных из некоторого класса ее решение существует, единственно и устойчиво по исходным данным.

Рассмотренная выше неустойчивая задача является некорректно поставленной. Применять для решения таких задач численные методы, как правило, нецелесообразно, поскольку возникающие в расчетах погрешности округления будут сильно возрастать в ходе вычислений, что приведет к значительному искажению результатов.

Вместе с тем отметим, что в настоящее время развиты методы решения некоторых некорректных задач. Это в основном так называемые *методы регуляризации*. Они основываются на замене исходной задачи корректно поставленной задачей. Последняя содержит некоторый параметр, при стремлении которого к нулю решение этой задачи переходит в решение исходной задачи.

3. Неустойчивость методов. Иногда при решении корректно поставленной задачи может оказаться неустойчивым метод ее решения. Такие случаи имели место в § 2. В частности, по этой причине при вычислении синуса большого аргумента был получен результат, не имеющий смысла. Рассмотрим еще один пример неустойчивого алгоритма. Построим численный метод вычисления интеграла

$$I_n = \int_0^1 x^n e^{x-1} dx, \quad n = 1, 2, \ldots$$

Интегрируя по частям, находим

$$I_1 = \int_0^1 x e^{x-1} dx = x e^{x-1}\Big|_0^1 - \int_0^1 e^{x-1} dx = \frac{1}{e},$$

$$I_2 = \int_0^1 x^2 e^{x-1} dx = x^2 e^{x-1}\Big|_0^1 - 2\int_0^1 x e^{x-1} dx = 1 - 2I_1,$$

. . . . . .

$$I_n = \int_0^1 x^n e^{x-1} dx = x^n e^{x-1}\Big|_0^1 - n\int_0^1 x^{n-1} e^{x-1} dx = 1 - nI_{n-1}.$$

Пользуясь полученным рекуррентным соотношением, вычисляем с двойной точностью (приводим результат, округленный до трех значащих цифр)

$$I_1 = 0.368, \qquad I_5 = 0.146,$$
$$I_2 = 0.264, \qquad \ldots$$
$$I_3 = 0.207, \qquad I_{17} = 0.0558,$$
$$I_4 = 0.171, \qquad I_{18} = -0.00369.$$

Значение интеграла I_{18} не может быть отрицательным, поскольку подынтегральная функция $x^{18}e^{x-1}$ на всем отрезке интегрирования $[0, 1]$ неотрицательна. Исследуем источник погрешности. Максимальная абсолютная погрешность при вычислении I_1 равна $0.5 \cdot 2^{-53} \approx 5 \cdot 10^{-17}$. Однако на каждом этапе эта погрешность умножается на число, модуль которого больше единицы $(-2, -3, \ldots, -18)$, что в итоге дает $18! \approx \approx 6.4 \cdot 10^{15}$. Это и приводит к результату, не имеющему смысла. Здесь

снова причиной накопления погрешностей является алгоритм решения задачи, который оказался неустойчивым.

Численный алгоритм (метод) называется *корректным* в случае существования и единственности численного решения при любых значениях исходных данных, а также в случае устойчивости этого решения относительно погрешностей исходных данных.

4. Понятие сходимости. При анализе точности вычислительного процесса одним из важнейших критериев является *сходимость* численного метода. Она означает близость получаемого численного решения задачи к истинному решению. Строгие определения разных оценок близости могут быть даны лишь с привлечением аппарата функционального анализа. Здесь мы ограничимся некоторыми понятиями сходимости, необходимыми для понимания последующего материала.

Рассмотрим понятие *сходимости итерационного процесса*. Этот процесс состоит в том, что для решения некоторой задачи и нахождения искомого значения определяемого параметра (например, корня нелинейного уравнения) строится метод последовательных приближений. В результате многократного повторения этого процесса (или *итераций*) получаем последовательность значений $x_1, x_2, \ldots, x_n, \ldots$. Говорят, что эта последовательность сходится к точному решению $x = a$, если при неограниченном возрастании числа итераций предел этой последовательности существует и равен a: $\lim\limits_{n\to\infty} x_n = a$. В этом случае имеем сходящийся численный метод.

Другой подход к понятию сходимости используется в методах дискретизации. Эти методы заключаются в замене задачи с непрерывными параметрами на задачу, в которой значения функций вычисляются в фиксированных точках. Это относится, в частности, к численному интегрированию, решению дифференциальных уравнений и т. п. Здесь под *сходимостью метода* понимается стремление значений решения дискретной модели задачи к соответствующим значениям решения исходной задачи при стремлении к нулю параметра дискретизации (например, шага интегрирования).

При рассмотрении сходимости важными понятиями являются ее вид, порядок и другие характеристики. С общей точки зрения эти понятия рассматривать здесь нецелесообразно; к ним будем обращаться при изучении конкретных численных методов.

Таким образом, для получения решения задачи с необходимой точностью ее постановка должна быть корректной, а используемый численный метод должен обладать устойчивостью (корректностью) и сходимостью.

Упражнения

1. Представить числа 175.4, -3.169, -0.00874 в нормализованном виде.

2. Записать в форме с фиксированной точкой числа $0.312 \cdot 10^3$, $-0.70 \cdot 10^1$, $0.465 \cdot 10^{-2}$.

3*. Получить соотношения (1.1) и (1.2).

4. Указать максимально возможные абсолютные и относительные погрешности приближенных чисел 27, -14.0, 0.00173, $0.745 \cdot 10^{-4}$, $-0.245 \cdot 10^{4}$, $-0.8960 \cdot 10^{2}$.

5. Оценить погрешности величин x, y, заданных соотношениями

$$x = \frac{a^3\sqrt{b}}{c^2+1}, \qquad y = \frac{\sqrt[3]{a-b}}{a^2+b^2+c^2} + \frac{a}{c},$$

при $a \approx 32$, $b \approx 17$, $c \approx 3.7$.

6. Найти относительные погрешности при вычислении определителей

$$d_1 = \begin{vmatrix} 0.19 & -0.27 \\ 1.4 & 2.3 \end{vmatrix}, \qquad d_2 = \begin{vmatrix} 17.5 & 10.4 \\ 10.4 & 6.18 \end{vmatrix}.$$

7. Каковы относительные погрешности объема шара и площади поверхности сферы, если их радиус известен с точностью до $10\,\%$?

8. Убедиться, что результат вычисления суммы чисел 103, 0.704, 0.537, 15.2 с округлением до трех значащих цифр после каждого сложения зависит от порядка суммирования. В каком порядке следует суммировать эти числа?

9. Что произойдет, если с помощью алгоритма, представленного на рис. 1.1, попытаться решить квадратное уравнение с коэффициентами $a = 10^{180}$, $b = -2 \cdot 10^{180}$, $c = -10^{-180}$? Как можно преодолеть возникшие трудности?

ГЛАВА 2

АППРОКСИМАЦИЯ ФУНКЦИЙ

§ 1. Понятие о приближении функций

1. Постановка задачи. Пусть величина y является функцией аргумента x. Это означает, что любому значению x из области определения поставлено в соответствие значение y. Вместе с тем на практике часто неизвестна явная связь между y и x, т. е. невозможно записать эту связь в виде некоторой зависимости $y = f(x)$. Иногда даже известная зависимость $y = f(x)$ оказывается настолько громоздкой (например, содержит трудно вычисляемые выражения, сложные интегралы и т. п.), что ее использование в практических расчетах требует слишком много времени.

Наиболее распространенным и практически важным случаем, когда вид связи между параметрами x и y неизвестен, является задание этой связи в виде некоторой таблицы $\{x_i, y_i\}$. Это означает, что дискретному множеству значений аргумента $\{x_i\}$ поставлено в соответствие множество значений функции $\{y_i\}$ ($i = 0, 1, \ldots, n$). Эти значения — либо результаты расчетов, либо экспериментальные данные. На практике нам могут понадобиться значения величины y и в других точках, отличных от узлов x_i. Однако получить эти значения можно лишь путем очень сложных расчетов или проведением дорогостоящих экспериментов.

Таким образом, с точки зрения экономии времени и средств мы приходим к необходимости использования имеющихся табличных данных для приближенного вычисления искомого параметра y при любом значении (из некоторой области) определяющего параметра x, поскольку точная связь $y = f(x)$ не известна (либо нам нецелесообразно ею пользоваться).

Этой цели и служит задача о приближении (*аппроксимации*) функций: данную функцию $f(x)$ требуется приближенно заменить (*аппроксимировать*) некоторой функцией $\varphi(x)$ так, чтобы отклонение (в некотором смысле) $\varphi(x)$ от $f(x)$ в заданной области было наименьшим. Функция $\varphi(x)$ при этом называется *аппроксимирующей*.

Аппроксимация рассмотренного выше типа, при которой приближение строится на заданном дискретном множестве точек $\{x_i\}$, называется *точечной*. К ней относятся интерполирование, среднеквадратичное приближение и др. При построении приближения на непрерывном множестве точек (например, на отрезке) аппроксимация называется *непрерывной* (или *интегральной*). К непрерывной аппроксимации относится, например, равномерное приближение.

Для практики весьма важен случай аппроксимации функции многочленом

$$\varphi(x) = P_m(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m. \qquad (2.1)$$

В дальнейшем будем чаще всего рассматривать аппроксимацию такого рода. При этом коэффициенты a_j будут подбираться так, чтобы достичь наименьшего отклонения многочлена от данной функции.

Что касается самого понятия «малое отклонение», то оно будет уточнено в дальнейшем — при рассмотрении конкретных способов аппроксимации.

2. Точечная аппроксимация. Одним из основных типов точечной аппроксимации является *интерполирование*. Оно состоит в следующем: для данной функции $y = f(x)$ строим *интерполирующую функцию* $\varphi(x)$ (например, многочлен (2.1)), принимающую в заданных точках x_i, те же значения y_i, что и функция $f(x)$, т. е.

$$\varphi(x_i) = y_i, \quad i = 0, 1, \ldots, n. \tag{2.2}$$

При этом предполагается, что среди значений x_i нет одинаковых, т. е. $x_i \neq x_k$ при $i \neq k$. Точки x_i называются *узлами интерполяции*.

Таким образом, близость интерполирующей функции (рис. 2.1, сплошная линия) к заданной функции состоит в том, что их значения совпадают на заданной системе точек.

Интерполирующая функция $\varphi(x)$ может строиться сразу для всего рассматриваемого интервала изменения x или отдельно для разных частей этого интервала. В первом случае говорят о *глобальной интерполяции*, во втором — о *кусочной* (или *локальной*) *интерполяции*.

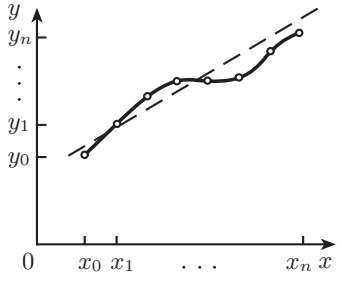


Рис. 2.1. Интерполяция и аппроксимация

Как правило, интерполирование используется для аппроксимации функции в промежуточных точках между крайними узлами интерполяции, т. е. при $x_0 < x < x_n$. Однако иногда оно применяется и для приближенного вычисления функции вне рассматриваемого отрезка $(x < x_0,\ x > x_n)$. Это приближение называют *экстраполяцией*.

Рассмотрим использование в качестве функции $\varphi(x)$ многочлена (2.1), называемого *интерполяционным многочленом*. При глобальной интерполяции, т. е. при построении одного многочлена для всего рассматриваемого интервала изменения x, для нахождения коэффициентов многочлена необходимо использовать все уравнения системы (2.2). Данная система содержит $n + 1$ уравнение, следовательно, с ее помощью можно определить $n + 1$ коэффициент. Поэтому максимальная степень интерполяционного многочлена $m = n$, и многочлен принимает вид

$$P_n(x) = a_0 + a_1 x + \ldots + a_n x^n. \tag{2.3}$$

Система уравнений (2.2) при использовании в качестве $\varphi(x)$ многочлена (2.3) является системой линейных алгебраических уравнений относительно неизвестных коэффициентов a_0, a_1, ..., a_n и имеет вид

$$\begin{aligned} a_0 + a_1 x_0 + \ldots + a_n x_0^n &= y_0, \\ a_0 + a_1 x_1 + \ldots + a_n x_1^n &= y_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_0 + a_1 x_n + \ldots + a_n x_n^n &= y_n. \end{aligned} \tag{2.4}$$

Определитель такой системы в линейной алгебре называется *определителем Вандермонда*. Можно показать, что определитель Вандермонда отличен от нуля, если $x_i \neq x_k$ при $i \neq k$, т. е. если среди узлов интерполяции нет совпадающих. Следовательно, в этом случае система (2.4) имеет единственное решение. Решив систему (2.4), можно построить интерполяционный многочлен. Такой метод построения интерполяционного многочлена называется *методом неопределенных коэффициентов*. Заметим вместе с тем, что этот метод требует значительного объема вычислений, особенно при большом числе узлов. Существуют более простые алгоритмы построения интерполяционных многочленов, которые будут рассмотрены в § 3.

Как видим, при интерполировании основным условием является прохождение графика интерполирующей функции через данные значения функции в узлах интерполяции. Однако в ряде случаев выполнение этого условия затруднительно или даже нецелесообразно.

Например, при большом количестве узлов интерполяции в случае глобальной интерполяции получается высокая степень многочлена (2.3). Кроме того, табличные данные могли быть получены путем измерений и содержать ошибки. Построение аппроксимирующей функции с условием обязательного прохождения ее графика через эти экспериментальные точки означало бы тщательное повторение допущенных при измерениях ошибок. Выход из этого положения может быть найден выбором такой функции, график которой проходит близко от данных точек (см. рис. 2.1, штриховая линия). Понятие «близко» уточняется при рассмотрении разных видов приближения.

Одним из таких видов является *среднеквадратичное приближение*. Если при этом используется многочлен (2.1), то $m \leqslant n$; случай $m = n$ соответствует глобальной интерполяции. На практике стараются подобрать аппроксимирующую функцию как можно более простого вида, например, многочлен степени $m = 1, 2, 3$.

Мерой отклонения функции $\varphi(x)$ от заданной функции $f(x)$ на множестве точек (x_i, y_i) $(i = 0, 1, \ldots, n)$ при среднеквадратичном приближении является величина S, равная сумме квадратов разностей между значениями аппроксимирующей и заданной функции в данных точках:

$$S = \sum_{i=0}^{n} [\varphi(x_i) - y_i]^2.$$

Аппроксимирующую функцию нужно подобрать так, чтобы величина S была наименьшей. В этом состоит *метод наименьших квадратов*, который будет рассмотрен в § 4.

3. Непрерывная аппроксимация. Равномерное приближение. Во многих случаях, особенно при обработке экспериментальных данных, среднеквадратичное приближение вполне приемлемо, поскольку оно сглаживает некоторые неточности функции $f(x)$ и дает достаточно правильное представление о ней. Иногда, однако, при построении приближения ставится более жесткое условие: требуется, чтобы во всех точках некоторого отрезка $[a, b]$ отклонение аппроксимирующей функции $\varphi(x)$ от функции $f(x)$ было по абсолютной величине меньшим заданной величины $\varepsilon > 0$:

$$|f(x) - \varphi(x)| < \varepsilon, \quad a \leqslant x \leqslant b.$$

В этом случае говорят, что функция $\varphi(x)$ *равномерно приближает* (*аппроксимирует*) функцию $f(x)$ с точностью ε на отрезке $[a, b]$.

Понятие равномерного приближения предполагает сравнение заданной и аппроксимирующей функций на непрерывном множестве — отрезке $[a, b]$. Поэтому равномерное приближение относится к непрерывной аппроксимации.

Введем понятие *абсолютного отклонения* Δ функции $\varphi(x)$ от функции $f(x)$ на отрезке $[a, b]$. Оно равно максимальному значению абсолютной величины разности между ними на данном отрезке:

$$\Delta = \max_{a \leqslant x \leqslant b} |f(x) - \varphi(x)|. \tag{2.5}$$

По аналогии можно ввести понятие *среднеквадратичного отклонения* $\bar{\Delta} = \sqrt{S/n}$ при среднеквадратичном приближении функций. На рис. 2.2 показано принципиальное различие двух рассматриваемых приближений.

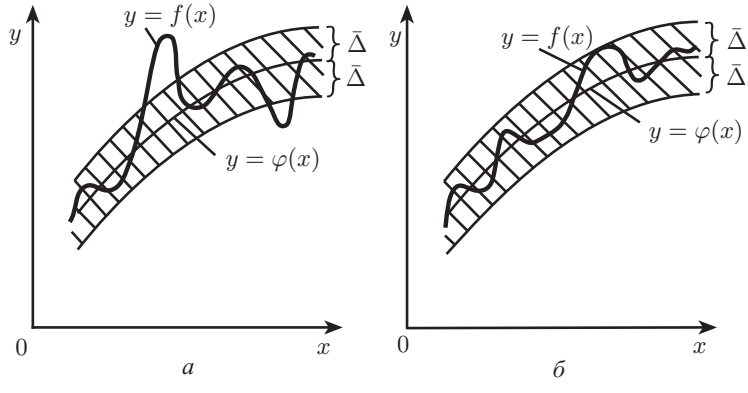


Рис. 2.2. Приближения: среднеквадратичное (*а*); равномерное (*б*)

Возможность построения многочлена, равномерно приближающего данную функцию, следует из *теоремы Вейерштрасса* об аппроксимации.

Т е о р е м а. *Если функция* $f(x)$ *непрерывна на отрезке* $[a,b]$, *то для любого* $\varepsilon > 0$ *существует многочлен* $P_m(x)$ *степени* $m = m(\varepsilon)$, *абсолютное отклонение которого от функции* $f(x)$ *на отрезке* $[a,b]$ *меньше* ε.

В частности, если функция $f(x)$ на отрезке $[a,b]$ разлагается в равномерно сходящийся степенной ряд, то в качестве аппроксимирующего многочлена можно взять частичную сумму этого ряда. Такой подход широко используется, например, при вычислении на компьютере значений элементарных функций.

Существует также понятие *наилучшего приближения функции* $f(x)$ многочленом $P_m(x)$ фиксированной степени m. В этом случае коэффициенты многочлена (2.1) следует выбрать так, чтобы на заданном отрезке $[a,b]$ величина абсолютного отклонения (2.5) была минимальной. Такой многочлен $P_m(x)$ называется *многочленом наилучшего равномерного приближения*. Существование и единственность многочлена наилучшего равномерного приближения вытекает из следующей теоремы.

Т е о р е м а. *Для любой функции* $f(x)$, *непрерывной на замкнутом ограниченном множестве* G, *и любого целого* $m \geqslant 0$ *существует многочлен* $P_m(x)$ *степени не выше* m, *абсолютное отклонение которого от функции* $f(x)$ *среди всех многочленов степени не выше* m *минимально, т. е.* $\Delta = \Delta_{\min}$, *причем такой многочлен единственный.*

Множество G обычно представляет собой некоторый отрезок $[a,b]$.

4. Вычисление многочленов. При аппроксимации функций, а также в некоторых других задачах приходится вычислять значения многочленов вида

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n. \tag{2.6}$$

Если проводить вычисления «в лоб», т. е. находить значения каждого члена и суммировать их, то при больших n потребуется выполнить большое число операций ($n^2 + n/2$ умножений и n сложений). Кроме того, это может привести к потере точности за счет погрешностей округления. В некоторых частных случаях удается выразить каждый последующий член через предыдущий и таким образом значительно сократить объем вычислений.

Анализ многочлена (2.6) в общем случае приводит к тому, что для исключения возведения x в степень в каждом члене многочлен целесообразно переписать в виде

$$P_n(x) = a_0 + x(a_1 + x(a_2 + \ldots + x(a_{n-1} + x a_n)\ldots).$$

Прием, с помощью которого многочлен представляется в таком виде, называется *схемой Горнера*. Соответствующий ему алгоритм вычисления

значения многочлена изображен на рис. 2.3. Этот метод требует n умножений и n сложений. Использование схемы Горнера для вычисления значений многочленов не только экономит машинное время, но и повышает точность вычислений за счет уменьшения погрешностей округления.

§ 2. Использование рядов

1. Элементарные функции. Как правило, при решении различных задач приходится вычислять значения элементарных функций (тригонометрических, показательных, логарифмических и др.). При ручном счете для этой цели могут быть использованы таблицы. Однако в вычислениях на компьютере ввод таблиц функций в машину потребовал бы больших затрат памяти. Кроме того, поиск нужного значения функции в памяти компьютера не простое для машины занятие. Поэтому для вычисления значений функций на компьютере используются разложения этих функций в степенные ряды. Например, функция $\sin x$ вычисляется с помощью ряда

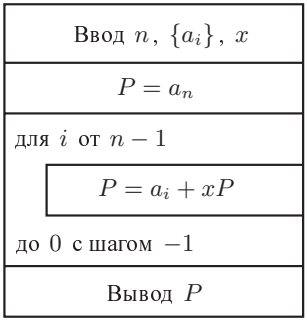


Рис. 2.3. Схема Горнера

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \qquad (2.7)$$

При известном значении аргумента x значение функции может быть получено с точностью до погрешностей округления. Количество используемых членов ряда (2.7) зависит от значения аргумента. Напомним, что в соответствии с правилами приближенных вычислений (см. гл. 1) для предотвращения влияния погрешностей округления необходимо выполнение неравенства $|x| < 1$.

С помощью степенных рядов вычисляются значения и других элементарных функций. В частности, для вычисления значений функции $\cos x$ можно использовать ряд (2.7) с учетом соотношения $\cos x = \sin(\pi/2 + x)$, а можно и непосредственно воспользоваться разложением в ряд функции $\cos x$:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \qquad (2.8)$$

Гиперболические синус и косинус можно вычислить с помощью разложения в ряд экспоненты e^x, поскольку

$$\operatorname{sh} x = (e^x - e^{-x})/2, \quad \operatorname{ch} x = (e^x + e^{-x})/2.$$

Можно также воспользоваться разложением в ряд самих функций $\operatorname{sh} x$ и $\operatorname{ch} x$. Так, при вычислении $\operatorname{sh} x$ для $x \approx 0$ в целях предотвращения потери точности из-за вычитания близких величин полезно использовать ряд

$$\operatorname{sh} x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots$$

Для вычисления на компьютере логарифмических функций достаточно иметь программу вычисления логарифма по одному основанию, например натурального логарифма. Для вычисления логарифма по другому основанию можно воспользоваться соотношением $\log_a x = \ln x / \ln a$.

В качестве примера построим алгоритм вычисления синуса с помощью ряда (2.7). Будем учитывать члены ряда, которые по абсолютной величине больше некоторого малого числа $\varepsilon > 0$, характеризующего точность вычисления. На практике, когда используют стандартные программы для вычисления функций, точность не задается. В этом случае учитываются все члены, большие машинного нуля, а точность результата определяется погрешностями округлений.

Возможный вариант алгоритма вычисления синуса с помощью ряда (2.7) изображен на рис 2.4 в виде структурограммы. Дадим некоторые пояснения к ней.

В алгоритме предусматривается выход из программы при малом значении аргумента, поскольку в этом случае $\sin x \approx x$. Выделяется абсолютная величина аргумента с учетом соотношений

$$x = k|x|, \quad \sin x = k \sin |x|, \quad k = \operatorname{sign} x.$$

Далее полагается, что $x > 0$.

При анализе точности вычислений отмечалось, что при суммировании ряда погрешность значительно меньше, если $|x| < 1$. Поэтому в структурограмме аргумент должен удовлетворять неравенству $x < \pi/4$. Это достигается последовательным уменьшением аргумента до значений $x < 2\pi$, $x < \pi$, $x < \pi/2$. Для этой цели использована функция $E(x)$, вычисляющая целую часть аргумента, а также формулы приведения $\sin(\pi \pm x) = \mp \mp \sin x$, $\sin(\pi/2 - x) = \cos x$. Например, при $x = 7.6\,\pi$ $(k = 1)$ получим следующий алгоритм:

$$n = E\left(\frac{7.6\,\pi}{2\pi}\right) = E(3.8) = 3,$$
$$x - 2\pi n = 7.6\,\pi - 6\pi = 1.6\,\pi > \pi, \quad k = -1,$$
$$x - \pi = 0.6\,\pi > \frac{\pi}{2}, \quad \pi - x = 0.4\,\pi > \frac{\pi}{4}, \quad \frac{\pi}{2} - x = 0.1\,\pi.$$

Текущее значение члена ряда на рис. 2.4 обозначено через u, значение функции — через y.

Здесь первые два члена ряда вычисляются непосредственно, а каждый последующий выражается через предшествующий. Например,

$$u_1 = x, \quad u_2 = -\frac{x^3}{3!}, \quad u_3 = \frac{x^5}{5!} = -u_2\,\frac{x^2}{4 \cdot 5}.$$

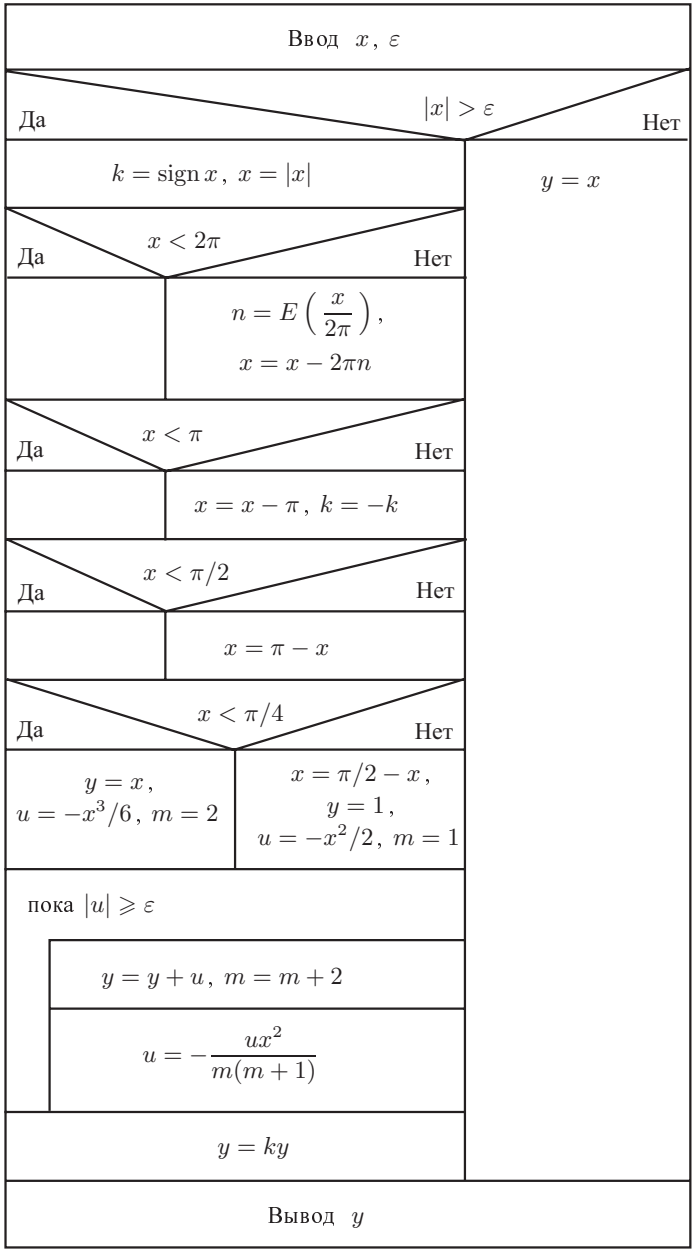


Рис. 2.4. Алгоритм вычисления синуса

Если $\pi/4 < x < \pi/2$, то проводится уменьшение аргумента до величины $\pi/2 - x$ и вычисление синуса сводится к вычислению косинуса, т. е. используется ряд (2.8). Для этого полагаем первоначально $y = 1$, $u = 1$, $m = 1$. В остальном алгоритм не меняется.

В рассмотренном алгоритме аргумент предполагается заданным в радианах. Если он задан в градусах, то следует предусмотреть перевод его в радианы, т. е. умножение на величину $\pi/180$.

Погрешность функции $y = \sin x$, полученной с помощью ряда (2.7) с использованием приведенного на рис. 2.4 алгоритма, состоит из двух частей — погрешности округления и *погрешности ограничения*, возникающей из-за учета лишь ограниченного числа членов ряда.

Погрешности ограничений при фиксированном числе членов ряда зависят от значения аргумента. При $|x| < \pi/4$ они весьма малы и сравнимы с погрешностями округлений даже при небольшом числе членов ряда, а с увеличением x возрастают. В частности, если ограничиться первыми четырьмя членами разложения (2.7) и провести вычисления с одинарной точностью, то погрешность при $x = \pi/4$ составит около $3 \cdot 10^{-7}$, а при $x = \pi/2$ — уже около $1.6 \cdot 10^{-4}$ (порядка первого отброшенного члена — в соответствии с признаком Лейбница).

2. Многочлены Чебышева. Из приведенного выше примера вычисления синуса с помощью ряда следует, что погрешности могут быть распределены неравномерно по рассматриваемому интервалу изменения аргумента. Одним из способов совершенствования алгоритма вычислений, позволяющих более равномерно распределить погрешность по всему интервалу, является использование многочленов Чебышева.

Многочлен Чебышева $T_n(x)$ степени n определяется следующей формулой:

$$T_n(x) = \frac{1}{2}\big[(x + \sqrt{1-x^2})^n + (x - \sqrt{1-x^2})^n\big], \quad -1 \leqslant x \leqslant 1, \quad n = 0, 1, \ldots \tag{2.9}$$

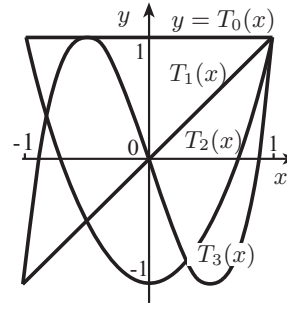


Рис. 2.5. Многочлены Чебышева

Легко показать, что (2.9) на самом деле является многочленом. Действительно, при возведении двучленов в степень n выражения $\sqrt{1-x^2}$ будут возведены в четные и нечетные степени. Эти выражения в четных степенях станут рациональными, а в каждой из нечетных степеней они будут присутствовать дважды с коэффициентами противоположных знаков, вследствие чего взаимно уничтожатся.

Приведем многочлены Чебышева, полученные по формуле (2.9) при $n = 0, 1, 2, 3$ (рис. 2.5):

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_2(x) = 2x^2 - 1, \quad T_3(x) = 4x^3 - 3x.$$

Для вычисления многочленов Чебышева можно воспользоваться следующим рекуррентным соотношением:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \ldots \tag{2.10}$$

В ряде случаев важно знать коэффициент a_n при старшем члене многочлена Чебышева степени n

$$T_n(x) = a_0 + a_1 x + \ldots + a_n x^n.$$

Разделив этот многочлен на x^n, найдем

$$a_n = \frac{T_n(x)}{x^n} - \frac{a_0}{x^n} - \ldots - \frac{a_{n-1}}{x}. \tag{2.11}$$

Определим многочлен Чебышева при $|x| > 1$ аналогично (2.9) с заменой $1 - x^2$ на $x^2 - 1$ и воспользуемся этим определением, переходя к пределу при $x \to \infty$ в (2.11). Получим

$$a_n = \lim_{x\to\infty} \frac{T_n(x)}{x^n} = \frac{1}{2}\left[\left(1 + \sqrt{1 - \frac{1}{x^2}}\right)^n + \left(1 - \sqrt{1 - \frac{1}{x^2}}\right)^n\right] = 2^{n-1}. \tag{2.12}$$

Многочлены Чебышева можно также представить в тригонометрической форме:

$$T_n(x) = \cos(n \arccos x), \quad n = 0, 1, \ldots$$

С помощью этих выражений могут быть получены формулы (2.9), (2.10).

Нули (*корни*) *многочленов Чебышева* на отрезке $[-1, 1]$ определяются формулой

$$z_k = \cos\frac{2k-1}{2n}\pi, \quad k = 1, 2, \ldots, n.$$

Они расположены неравномерно на отрезке и сгущаются к его концам.

Вычисляя экстремумы многочлена Чебышева по обычным правилам (с помощью производных), можно найти его максимумы и минимумы:

$$x_k = \cos(k\pi/n), \quad k = 1, 2, \ldots, n-1.$$

В этих точках многочлен принимает поочередно значения $T_n(x_k) = \pm 1$, т. е. все максимумы равны 1, а минимумы равны -1. На границах отрезка значения многочленов Чебышева равны ± 1.

Многочлены Чебышева широко используются при аппроксимации функций. Рассмотрим их применение для улучшения приближения функций с помощью степенных рядов, а именно для более равномерного распределения погрешностей аппроксимации (2.7) по заданному отрезку $[-\pi/2, \pi/2]$.

Отрезок $[-\pi/2, \pi/2]$ является не совсем удобным при использовании многочленов Чебышева, поскольку они обычно рассматриваются на

стандартном отрезке $[-1,1]$. Первый отрезок легко привести ко второму заменой переменной x на $\pi x/2$. В этом случае ряд (2.7) для аппроксимации синуса на отрезке $[-1,1]$ примет вид

$$\sin\frac{\pi x}{2}=\frac{\pi x}{2}-\frac{1}{3!}\left(\frac{\pi x}{2}\right)^3+\frac{1}{5!}\left(\frac{\pi x}{2}\right)^5-\dots \tag{2.13}$$

При использовании этого ряда погрешность вычисления функции в окрестности концов отрезка $x=\pm1$ существенно возрастает и становится значительно больше, чем в окрестности точки $x=0$. Если вместо (2.13) использовать ряд

$$\sin\frac{\pi x}{2}=c_0+c_1T_1(x)+c_2T_2(x)+\dots, \tag{2.14}$$

членами которого являются многочлены Чебышева, то погрешность будет распределена равномерно по всему отрезку (рис. 2.6). В частности, при использовании многочленов Чебышева до седьмой степени включительно (т. е. при использовании четырех ненулевых членов (2.14)) погрешность находится в интервале $(-5.9\div5.9)\cdot10^{-7}$. Для сравнения напомним, что при использовании четырех членов ряда Тейлора погрешность для этой задачи на концах отрезка составляет $\pm1.6\cdot10^{-4}$.

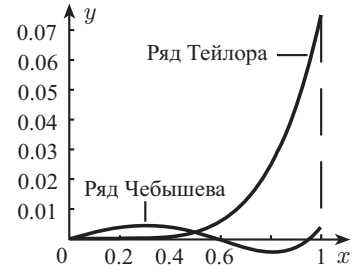


Рис. 2.6. Погрешность аппроксимации функции $y=\sin(\pi x/2)$ с помощью рядов Тейлора и Чебышева с учетом членов до x^3 включительно

Для нахождения коэффициентов ряда Чебышева нужно воспользоваться свойством *ортогональности* многочленов Чебышева на отрезке $[-1,1]$ *с весом* $1/\sqrt{1-x^2}$:

$$\int_{-1}^{1}\frac{T_m(x)T_n(x)}{\sqrt{1-x^2}}\,dx=\begin{cases}0, & m\neq n,\\ \dfrac{\pi}{2}, & m=n>0,\\ \pi, & m=n=0.\end{cases} \tag{2.15}$$

Если записать ряд Чебышева для функции $f(x)$ в виде

$$f(x)=\frac{c_0}{2}+c_1T_1(x)+c_2T_2(x)+\dots, \tag{2.16}$$

умножить обе части равенства на $T_n(x)/\sqrt{1-x^2}$, проинтегрировать ряд почленно по отрезку $[-1,1]$ и воспользоваться (2.15), то получим

$$c_n=\frac{2}{\pi}\int_{-1}^{1}\frac{f(x)T_n(x)}{\sqrt{1-x^2}}\,dx,\quad n=0,1,2,\dots \tag{2.17}$$

Заметим, что коэффициенты ряда Чебышева находятся аналогично тому, как находятся коэффициенты известного из курса высшей математики тригонометрического ряда Фурье.

Входящие в (2.17) интегралы в общем случае можно найти только численно. На практике часто находят коэффициенты ряда Чебышева приближенно, заменяя функцию $f(x)$ частичной суммой ее ряда Тейлора, содержащей достаточное для требуемой точности количество членов ряда. Рассмотрим этот подход более подробно.

Пусть частичная сумма ряда Тейлора, представленная в виде многочлена, используется для приближения функции $f(x)$ на стандартном отрезке $[-1, 1]$, т. е.

$$f(x) \approx \varphi(x) = a_0 + a_1 x + \ldots + a_n x^n. \tag{2.18}$$

Если рассматриваемый отрезок $[a, b]$ отличается от стандартного, то его всегда можно привести к стандартному заменой переменной

$$t = \frac{a+b}{2} + \frac{b-a}{2} x, \quad -1 \leqslant x \leqslant 1.$$

Каждая степень x^m может быть выражена через многочлены Чебышева степеней, не превышающих m (в приложении Б приведены соответствующие формулы). В результате ряд Чебышева для функции $\varphi(x)$ будет являться конечной суммой:

$$\varphi(x) = \frac{c_0^*}{2} + c_1^* T_1(x) + \ldots + c_{n-1}^* T_{n-1}(x) + c_n^* T_n(x), \tag{2.19}$$

где c_0^*, c_1^*, $\ldots$, c_n^* — приближенные значения коэффициентов c_0, c_1, $\ldots$, c_n в (2.16). Если в такой сумме привести подобные слагаемые, содержащие x в одинаковых степенях, т. е. явно представить ее в виде многочлена, то этот многочлен совпадет с многочленом (2.18).

Поскольку, как нам известно, погрешность при использовании ряда Чебышева меньше, чем при использовании ряда Тейлора, требуемую точность можно сохранить и при меньшем количестве слагаемых в (2.19). Несколько последних из них можно отбросить.

Рассмотрим, что происходит при отбрасывании слагаемого с T_n. Многочлен $T_n(x)$ входит в частичной сумме (2.18) только в выражение для x^n. Поэтому чтобы найти многочлен, получающийся после отбрасывания слагаемого с T_n в (2.19), не нужно вычислять коэффициенты c_0^*, c_1^*, $\ldots$, c_{n-1}^*. Достаточно выразить x^n через $T_n(x)$ и меньшие степени x, подставить полученное выражение в (2.18), а затем отбросить в (2.18) слагаемое с T_n.

Оценим допускаемую при этом погрешность. Из (2.12) следует, что многочлен Чебышева $T_n(x)$ можно записать в виде

$$T_n(x) = b_0 + b_1 x + b_2 x^2 + \ldots + 2^{n-1} x^n.$$

Отсюда получаем

$$x^n = -2^{1-n}(b_0 + b_1 x + \ldots + b_{n-1}x^{n-1}) + 2^{1-n}T_n(x). \qquad (2.20)$$

Если отбросить последний член, то допущенную при этом погрешность Δ_1 легко оценить: $|\Delta_1| \leqslant 2^{1-n}$, поскольку $|T_n(x)| \leqslant 1$. Тогда погрешность Δ, допущенная в (2.18) и (2.19), будет оцениваться как $|\Delta| \leqslant |a_n|2^{1-n}$.

Из (2.20) получаем, что x^n есть с точностью до Δ_1 линейная комбинация более низких степеней x. Подставляя эту линейную комбинацию в (2.18), приходим к многочлену степени $n-1$ вместо многолена степени n. Этот процесс может быть продолжен до тех пор, пока погрешность не превышает допустимого значения.

Описанную процедуру можно трактовать как повышение точности аппроксимации функции с помощью ряда Тейлора. Используем эту процедуру для повышения точности аппроксимации (2.13). Поставим задачу примерно сохранить погрешность $5.9 \cdot 10^{-7}$, имевшую место при использовании многочленов Чебышева до седьмой степени. Будем учитывать члены ряда (2.13) до 11-й степени включительно. При этом допускается погрешность примерно $6 \cdot 10^{-8}$. Вычисляя коэффициенты при степенях x, получаем, округляя до восьми разрядов,

$$\sin(\pi x/2) \approx 1.5707963\,x - 0.64596410\,x^3 + 0.079692626\,x^5 - \\ - 0.0046817541\,x^7 + 0.00016044118\,x^9 + 0.0000035988432\,x^{11}. \qquad (2.21)$$

Многочлен Чебышева 11-й степени имеет вид

$$T_{11} = 1024\,x^{11} - 2816\,x^9 + 2816\,x^7 - 1232\,x^5 + 220\,x^3 - 11x.$$

Выразим отсюда x^{11} через более низкие степени:

$$x^{11} = 2^{-10}(11x - 220\,x^3 + 1232\,x^5 - 2816\,x^7 + 2816\,x^9 + T_{11}).$$

Подставляя в (2.21) вместо x^{11} правую часть этого равенства и вычисляя новые значения коэффициентов, получаем

$$\sin(\pi x/2) \approx 1.5707963\,x - 0.64596332\,x^3 + 0.079688296\,x^5 - \\ - 0.0046718573\,x^7 + 0.00015054437\,x^9 - 0.00000000351\,T_{11}. \qquad (2.22)$$

Отбрасывая последний член этого разложения, мы допускаем погрешность $|\Delta| \leqslant 3.51 \cdot 10^{-9}$. Из-за приближенного вычисления коэффициентов при степенях x и погрешности ограничения реальная погрешность больше. Она составляет $4 \cdot 10^{-8}$, что все равно значительно меньше чем $5.9 \cdot 10^{-7}$. Поэтому процедуру отбрасывания слагаемого можно повторить.

Подставляя в (2.22) (без последнего члена с T_{11}) выражение x^9 через более низкие степени

$$x^9 = 2^{-8}(-9x + 120\,x^3 - 432\,x^5 + 576\,x^7 + T_9),$$

получаем

$$\sin(\pi x/2) \approx 1.5707910\,x - 0.64589276\,x^3 + 0.079434253\,x^5 - \\ - 0.0043331325\,x^7 + 0.000000588\,T_9. \quad (2.23)$$

Отбрасывая последний член этого разложения, мы допускаем погрешность $|\Delta| \leqslant 5.88 \cdot 10^{-7}$.

Суммарная погрешность (2.23) без последнего члена с T_9 составит $6.4 \cdot 10^{-7}$, что лишь немногим больше погрешности $5.9 \cdot 10^{-7}$, допускаемой при вычислении коэффициентов ряда Чебышева по (2.17).

В приложении Б приведены некоторые формулы, необходимые при использовании многочленов Чебышева.

3. Рациональные приближения. Рассмотрим другой вид аппроксимации функций — с помощью дробно-рационального выражения. Функцию представим в виде отношения двух многочленов некоторой степени. Пусть, например, это будут многочлены третьей степени, т. е. представим функцию $f(x)$ в виде дробно-рационального выражения

$$f(x) = \frac{b_0 + b_1x + b_2x^2 + b_3x^3}{1 + c_1x + c_2x^2 + c_3x^3}. \quad (2.24)$$

Значение свободного члена в знаменателе $c_0 = 1$ не нарушает общности этого выражения, поскольку при $c_0 \neq 1$ числитель и знаменатель можно разделить на c_0. Если же $c_0 = 0$, то $f(x)$ будет иметь особенность при $x = 0$. Такую аппроксимацию рассматривать не будем.

Перепишем выражение (2.24) в виде

$$b_0 + b_1x + b_2x^2 + b_3x^3 = (1 + c_1x + c_2x^2 + c_3x^3)f(x).$$

Используя разложение функции $f(x)$ в ряд Тейлора

$$f(x) = a_0 + a_1x + a_2x^2 + \ldots \quad (2.25)$$

и учитывая члены до шестой степени включительно, получаем

$$b_0 + b_1x + b_2x^2 + b_3x^3 = (1 + c_1x + c_2x^2 + c_3x^3) \times \\ \times (a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6).$$

Преобразуем правую часть, этого равенства, записав ее разложение по степеням x:

$$b_0 + b_1x + b_2x^2 + b_3x^3 = a_0 + x(a_1 + a_0c_1) + x^2(a_2 + a_1c_1 + a_0c_2) + \\ + x^3(a_3 + a_2c_1 + a_1c_2 + a_0c_3) + x^4(a_4 + a_3c_1 + a_2c_2 + a_1c_3) + \\ + x^5(a_5 + a_4c_1 + a_3c_2 + a_2c_3) + x^6(a_6 + a_5c_1 + a_4c_2 + a_3c_3).$$

Приравнивая коэффициенты при одинаковых степенях x в левой и правой частях, получаем следующую систему уравнений:

$$\begin{aligned} b_0 &= a_0, \\ b_1 &= a_1 + a_0c_1, \\ b_2 &= a_2 + a_1c_1 + a_0c_2, \\ b_3 &= a_3 + a_2c_1 + a_1c_2 + a_0c_3, \\ 0 &= a_4 + a_3c_1 + a_2c_2 + a_1c_3, \\ 0 &= a_5 + a_4c_1 + a_3c_2 + a_2c_3, \\ 0 &= a_6 + a_5c_1 + a_4c_2 + a_3c_3. \end{aligned} \tag{2.26}$$

Решив эту систему, найдем коэффициенты b_0, b_1, b_2, b_3, c_1, c_2, c_3 необходимые для аппроксимации (2.24).

Пример. Рассмотрим рациональное приближение для функции $f(x) = \sin(\pi x/2)$. Воспользуемся представлением (2.24), которое в данном случае упрощается, поскольку функция $\sin x$ нечетная. В частности, в числителе можем оставить только члены с нечетными степенями x, а в знаменателе — с четными; коэффициенты при других степенях x равны нулю: $b_0 = b_2 = c_1 = c_3 = 0$.

Коэффициенты b_1, b_3, c_2 найдем из системы уравнений (2.26), причем значения коэффициентов a_0, a_1, … , a_6 разложения функции в ряд Тейлора (2.25) можем взять из выражения (2.13), т. е.

$$a_0 = a_2 = a_4 = a_6 = 0, \quad a_1 = \frac{\pi}{2}, \quad a_3 = -\frac{\pi^3}{8 \cdot 3!}, \quad a_5 = \frac{\pi^5}{32 \cdot 5!}.$$

Система уравнений (2.26) в данном случае примет вид

$$\begin{aligned} b_1 &= \frac{\pi}{2}, \\ b_3 &= -\frac{\pi^3}{8 \cdot 3!} + \frac{\pi}{2}\, c_2, \\ 0 &= \frac{\pi^5}{32 \cdot 5!} - \frac{\pi^3}{8 \cdot 3!}\, c_2. \end{aligned}$$

Отсюда находим $b_1 = \pi/2$, $b_3 = -7\pi^3/480$, $c_2 = \pi^2/80$.

Таким образом, дробно-рациональное приближение (2.24) для функции $\sin(\pi x/2)$ примет вид

$$\sin\frac{\pi x}{2} = \frac{(\pi/2)x - (7\pi^3/480)x^3}{1 + (\pi^2/80x^2)}. \tag{2.27}$$

Это приближение по точности равносильно аппроксимации (2.13) с учетом членов до пятого порядка включительно.

На практике с целью экономии числа операций выражение (2.24) представляется в виде *цепной дроби*. Представим в таком виде дробно-рациональное выражение (2.27). Сначала перепишем это выражение, вынося

за скобки коэффициенты при x^3 и x^2. Получим

$$\sin\frac{\pi x}{2} = -\frac{7\pi}{6}\,\frac{x^3 - (60/7)(2/\pi)^2 x}{x^2 + 20(2/\pi)^2}.$$

Разделим числитель на знаменатель по правилу деления многочленов и введем обозначения для коэффициентов. Получим

$$\sin\frac{\pi x}{2} = k_1\left(x + \frac{k_2 x}{x^2 + k_3}\right),$$

$$k_1 = -\frac{7\pi}{6}, \quad k_2 = -\frac{200}{7}\left(\frac{2}{\pi}\right)^2, \quad k_3 = 20\left(\frac{2}{\pi}\right)^2.$$

Полученное выражение можно записать в виде

$$\sin\frac{\pi x}{2} = k_1\left(x + \frac{k_2}{x + k_3/x}\right).$$

Для вычисления значения функции по этой формуле требуется меньше операций (два деления, два сложения, одно умножение), чем для вычисления с помощью выражения (2.27) или усеченного ряда Тейлора (2.13) с использованием схемы Горнера (четыре умножения, два сложения). Следует, однако, иметь в виду, что процессору на выполнение операции деления обычно требуется гораздо больше времени, чем для выполнения операции умножения. Поэтому вопрос об использовании той или иной аппроксимирующей функции требует в каждом конкретном случае дополнительного исследования.

Приведем формулы для приближения некоторых элементарных функций с помощью цепных дробей, указывая интервалы изменения аргумента и погрешности Δ:

$$e^x = 1 + \cfrac{x}{-0.5x + \cfrac{k_0 + k_1 x^2}{1 + k_2 x^2}},$$

$$k_0 = 1.0000000020967, \quad k_1 = 0.0999743507186, \quad k_2 = 0.0166411490538,$$

$$-\frac{1}{2}\ln 2 \leqslant x \leqslant \frac{1}{2}, \quad |\Delta| \leqslant 10^{-10};$$

$$\ln(1+x) = k_0 + \cfrac{x}{k_1 + \cfrac{x}{k_2 + \cfrac{x}{k_3 + \cfrac{x}{k_4 + k_5 x}}}},$$

$$k_0 = 0.0000000894, \quad k_1 = 1.0000091365, \quad k_2 = 2.0005859000,$$
$$k_3 = 3.0311932666, \quad k_4 = 1.0787748225, \quad k_5 = 0.1124191908,$$

$$0 \leqslant x \leqslant 1, \quad |\Delta| \leqslant 10^{-7};$$

$$\operatorname{tg}\frac{\pi x}{4} = x\left(k_0 + \cfrac{x^2}{k_1 + \cfrac{x^2}{k_2 + k_3 x^2}}\right),$$

$$k_0 = 0.7853980289, \quad k_1 = 6.1922344479, \quad k_2 = -0.6545887679,$$
$$k_3 = 0.0020366541, \quad -1 \leqslant x \leqslant 1, \quad |\Delta| \leqslant 2 \cdot 10^{-8};$$

$$\operatorname{arctg} x = x\left(k_0 + \cfrac{x^2}{k_1 + \cfrac{x^2}{k_2 + \cfrac{x^2}{k_3 + k_4 x^2}}}\right),$$

$$k_0 = 0.99999752, \quad k_1 = -3.00064286, \quad k_2 = -0.55703890,$$
$$k_3 = -17.03715998, \quad k_4 = -4.86455143,$$
$$-1 \leqslant x \leqslant 1, \quad |\Delta| \leqslant 2 \cdot 10^{-7}.$$

§ 3. Интерполирование

1. Линейная и квадратичная интерполяции. Простейшим и часто используемым видом локальной интерполяции является *линейная* (или *кусочно линейная*) *интерполяция*. Она состоит в том, что заданные точки (x_i, y_i) ($i = 0, 1, \ldots, n$) соединяются прямолинейными отрезками, и функция $f(x)$ приближается ломаной с вершинами в данных точках.

Уравнения каждого отрезка ломаной в общем случае разные. Поскольку имеется n интервалов (x_{i-1}, x_i), то для каждого из них в качестве уравнения интерполяционного многочлена используется уравнение прямой, проходящей через две точки. В частности, для i-го интервала можно написать уравнение прямой, проходящей через точки (x_{i-1}, y_{i-1}) и (x_i, y_i), в виде

$$\frac{y - y_{i-1}}{y_i - y_{i-1}} = \frac{x - x_{i-1}}{x_i - x_{i-1}}.$$

Отсюда

$$y = a_i x + b_i, \quad x_{i-1} \leqslant x \leqslant x_i, \tag{2.28}$$
$$a_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}, \quad b_i = y_{i-1} - a_i x_{i-1}.$$

Следовательно, при использовании линейной интерполяции сначала нужно определить интервал, в который попадает значение аргумента x, а затем подставить его в формулу (2.28) и найти приближенное значение функции в этой точке. Данный алгоритм представлен на рис. 2.7. Попытайтесь разобраться, будет ли работать этот алгоритм, если окажется, что $x < x_0$ или $x > x_n$, и при необходимости модифицировать его.

Рассмотрим теперь случай *квадратичной интерполяции*. В качестве интерполяционной функции на отрезке $[x_{i-1}, x_{i+1}]$ принимается квадратный трехчлен. Такую интерполяцию называют также *параболической*.

Уравнение квадратного трехчлена

$$y = a_i x^2 + b_i x + c_i, \quad x_{i-1} \leqslant x_i \leqslant x_{i+1}, \tag{2.29}$$

содержит три неизвестных коэффициента a_i, b_i, c_i, для определения которых необходимы три уравнения. Ими служат условия прохождения параболы (2.29) через три точки (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}). Эти условия можно записать в виде

$$\begin{aligned} a_i x_{i-1}^2 + b_i x_{i-1} + c_i &= y_{i-1}, \\ a_i x_i^2 \ \ + b_i x_i \ \ \ + c_i &= y_i, \\ a_i x_{i+1}^2 + b_i x_{i+1} + c_i &= y_{i+1}. \end{aligned} \tag{2.30}$$

Если отвлечься от того, что отрезок $[x_{i-1}, x_{i+1}]$ является подмножеством отрезка $[x_0, x_n]$, и рассмотреть его отдельно, то квадратичную интерполяцию (2.29) можно трактовать как глобальную с $n = 2$, а систему (2.30) — как частный случай системы (2.4).

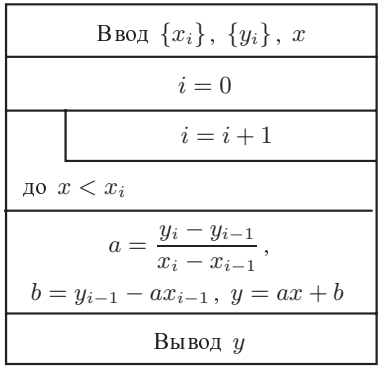


Рис. 2.7. Алгоритм линейной интерполяции

Алгоритм вычисления приближенного значения функции с помощью квадратичной интерполяции можно записать в виде структурограммы, как и для случая линейной интерполяции (см. рис. 2.7). Вместо формулы (2.28) нужно использовать (2.29) с учетом решения системы линейных уравнений (2.30). Интерполяция для любой точки $x \in [x_0, x_n]$ проводится по трем ближайшим к ней узлам.

Пример. Найти приближенное значение функции $y = f(x)$ при $x = 0.32$, если известна следующая таблица ее значений:

x	0.15	0.30	0.40	0.55
y	2.17	3.63	5.07	7.78

Воспользуемся сначала формулой линейной интерполяции (2.28). Значение $x = 0.32$ находится между узлами $x_{i-1} = 0.30$ и $x_i = 0.40$. В этом случае

$$a_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} = \frac{5.07 - 3.63}{0.40 - 0.30} = 14.4,$$
$$b_i = y_{i-1} - a_i x_{i-1} = 3.63 - 14.4 \cdot 0.30 = -0.69,$$
$$y \approx 14.4\,x - 0.69 = 14.4 \cdot 0.32 - 0.69 = 3.92.$$

Найдем теперь приближенное значение функции с помощью формулы квадратичной интерполяции (2.29). Составим систему уравнений (2.30)

с учетом ближайших к точке $x = 0.32$ узлов: $x_{i-1} = 0.15$, $x_i = 0.30$, $x_{i+1} = 0.40$. Соответственно $y_{i-1} = 2.17$, $y_i = 3.63$, $y_{i+1} = 5.07$. Система (2.30) запишется в виде

$$\begin{aligned} 0.15^2\, a_i \;+\; 0.15\, b_i \;+\; c_i \;&=\; 2.17,\\ 0.30^2\, a_i \;+\; 0.30\, b_i \;+\; c_i \;&=\; 3.63,\\ 0.40^2\, a_i \;+\; 0.40\, b_i \;+\; c_i \;&=\; 5.07. \end{aligned}$$

Решая эту систему, находим $a_i = 18.67$, $b_i = 1.33$, $c_i = 1.55$. Искомое значение функции $y \approx 18.67 \cdot 0.32^2 + 1.33 \cdot 0.32 + 1.55 \approx 3.89$.

2. Многочлен Лагранжа. Перейдем к случаю глобальной интерполяции, т. е. к построению интерполяционного многочлена, единого для всего отрезка $[x_0, x_n]$. Как уже отмечалось в § 1, для построения такого многочлена достаточно решить систему (2.4), однако этот путь неэффективен.

Будем искать интерполяционный многочлен в виде линейной комбинации многочленов степени n:

$$L(x) = y_0 l_0(x) + y_1 l_1(x) + \ldots + y_n l_n(x). \tag{2.31}$$

При этом потребуем, чтобы каждый многочлен $l_i(x)$ обращался в нуль во всех узлах интерполяции, за исключением одного (i-го), где он должен равняться единице. Легко проверить, что этим условиям при $i = 0$ отвечает многочлен вида

$$l_0(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)}. \tag{2.32}$$

Действительно, $l_0(x_0) = 1$. При $x = x_1, x_2, \ldots, x_n$ числитель выражения (2.32) обращается в нуль. По аналогии с (2.32) получим

$$\begin{aligned} l_1(x) &= \frac{(x - x_0)(x - x_2)\ldots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\ldots(x_1 - x_n)},\\ l_2(x) &= \frac{(x - x_0)(x - x_1)(x - x_3)\ldots(x - x_n)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)\ldots(x_3 - x_n)},\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ l_i(x) &= \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)},\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ l_n(x) &= \frac{(x - x_0)(x - x_1)\ldots(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)\ldots(x_n - x_{n-1})}. \end{aligned} \tag{2.33}$$

Подставляя в (2.31) выражения (2.32), (2.33), находим

$$L(x) = \sum_{i=0}^{n} y_i \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}. \tag{2.34}$$

Эта формула определяет *интерполяционный многочлен Лагранжа*.

Единственность представления многочлена (2.31) в виде (2.34) следует из единственности решения системы (2.4).

Из формулы (2.34) можно получить выражения для линейной ($n = 1$) и квадратичной ($n = 2$) интерполяций:

$$L(x) = \frac{x - x_1}{x_0 - x_1}\, y_0 + \frac{x - x_0}{x_1 - x_0}\, y_1, \quad n = 1;$$
$$L(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}\, y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}\, y_1 + \tag{2.35}$$
$$+ \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}\, y_2, \quad n = 2.$$

Существует несколько обобщений интерполяционного многочлена Лагранжа. Например, довольно широко используются *интерполяционные многочлены Эрмита*. Здесь наряду со значениями функции y_i в узлах x_i задаются значения ее производной y_i'. Задача состоит в том, чтобы найти многочлен $\varphi(x)$ степени $2n + 1$, значения которого и значения его производной в узлах x_i удовлетворяют соответственно соотношениям

$$\varphi(x_i) = y_i, \quad \varphi'(x_i) = y_i', \quad i = 0, 1, \ldots, n.$$

В этом случае также существует единственное решение, если все x_i различны.

3. Многочлен Ньютона. До сих пор не делалось никаких предположений о законе распределения узлов интерполяции. Теперь рассмотрим случай равноотстоящих значений аргумента, т. е. $x_i - x_{i-1} = h = \text{const}$ ($i = 1,\ 2,\ \ldots,\ n$). Величина h называется *шагом*.

Введем также понятие *конечных разностей*. Пусть известны значения функции в узлах x_i: $y_i = f(x_i)$. Составим разности значений функции:

$$\Delta y_0 = y_1 - y_0 = f(x_0 + h) - f(x_0),$$
$$\Delta y_1 = y_2 - y_1 = f(x_0 + 2h) - f(x_0 + h),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$\Delta y_{n-1} = y_n - y_{n-1} = f(x_0 + nh) - f(x_0 + (n-1)h).$$

Эти значения называются *первыми разностями* (или *разностями первого порядка*) функции.

Можно составить *вторые разности* функции:

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0, \quad \Delta^2 y_1 = \Delta y_2 - \Delta y_1, \quad \ldots$$

Аналогично составляются *разности порядка* k:

$$\Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i, \quad i = 0, 1, \ldots, n - 1.$$

Конечные разности можно выразить непосредственно через значения функции. Например,

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0 = (y_2 - y_1) - (y_1 - y_0) = y_2 - 2y_1 + y_0,$$
$$\Delta^3 y_0 = \Delta^2 y_1 - \Delta^2 y_0 = \ldots = y_3 - 3y_2 + 3y_1 - y_0.$$

Аналогично для любого k можно написать

$$\Delta^k y_0 = y_k - k y_{k-1} + \frac{k(k-1)}{2!}\, y_{k-2} + \ldots + (-1)^k y_0. \qquad (2.36)$$

Эту формулу можно записать и для значения разности в узле x_i:

$$\Delta^k y_i = y_{k+i} - k y_{k+i-1} + \frac{k(k-1)}{2!}\, y_{k+i-2} + \ldots + (-1)^k y_i.$$

Используя конечные разности, можно определить y_k:

$$y_k = y_0 + k\Delta y_0 + \frac{k(k-1)}{2!}\, \Delta^2 y_0 + \ldots + \Delta^k y_0.$$

Перейдем к построению интерполяционного многочлена Ньютона. Этот многочлен будем искать в следующем виде:

$$N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots \\ \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \qquad (2.37)$$

График многочлена должен проходить через заданные узлы, т. е. $N(x_i) = y_i$ $(i = 0, 1, \ldots, n)$. Эти условия используем для нахождения коэффициентов многочлена:

$$\begin{aligned} &N(x_0) = a_0 = y_0, \\ &N(x_1) = a_0 + a_1(x_1 - x_0) = a_0 + a_1 h = y_1, \\ &N(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) = \\ &\qquad = a_0 + 2a_1 h + 2a_2 h^2 = y_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{aligned}$$

Найдем отсюда коэффициенты a_0, a_1, a_2:

$$a_0 = y_0, \quad a_1 = \frac{y_1 - a_0}{h} = \frac{y_1 - y_0}{h} = \frac{\Delta y_0}{h},$$

$$a_2 = \frac{y_2 - a_0 - 2a_1 h}{2h^2} = \frac{y_2 - y_0 - 2\Delta y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2}.$$

Аналогично можно найти и другие коэффициенты. Общая формула имеет вид

$$a_k = \frac{\Delta^k y_0}{k!\, h^k}, \quad k = 0, 1, \ldots, n.$$

Подставляя эти выражения в формулу (2.37), получаем следующий вид *интерполяционного многочлена Ньютона*:

$$N(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2!\, h^2}(x - x_0)(x - x_1) + \ldots \\ \ldots + \frac{\Delta^n y_0}{n!\, h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \qquad (2.38)$$

Конечные разности $\Delta^k y_0$ могут быть вычислены по формуле (2.36).

Формулу (2.38) часто записывают в другом виде. Для этого вводится переменная $t = (x - x_0)/h$; тогда

$$x = x_0 + th, \quad \frac{x - x_1}{h} = \frac{x - x_0 - h}{h} = t - 1,$$
$$\frac{x - x_2}{h} = t - 2, \quad \ldots, \quad \frac{x - x_{n-1}}{h} = t - n + 1.$$

С учетом этих соотношений формулу (2.38) можно переписать в виде

$$N(x) = N(x_0 + th) = y_0 + t\,\Delta y_0 + \frac{t(t-1)}{2!}\,\Delta^2 y_0 + \ldots$$
$$\ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\,\Delta^n y_0. \quad (2.39)$$

Полученное выражение называется *первым интерполяционным многочленом Ньютона для интерполирования вперед*. Оно может аппроксимировать данную функцию $y = f(x)$ на всем отрезке изменения аргумента $[x_0, x_n]$. Однако с точки зрения повышения точности расчетов (путем уменьшения погрешностей округления) более целесообразно использовать (2.39) для вычисления значении функции в точках левой половины рассматриваемого отрезка.

Для правой половины отрезка $[x_0, x_n]$ разности лучше вычислять справа налево. В этом случае $t = (x - x_n)/h$, т. е. $t < 0$, и интерполяционный многочлен Ньютона можно получить в виде

$$N(x) = N(x_n + th) = y_n + t\,\Delta y_{n-1} + \frac{t(t+1)}{2!}\,\Delta^2 y_{n-2} + \ldots$$
$$\ldots + \frac{t(t+1)\ldots(t+n-1)}{n!}\,\Delta^n y_0. \quad (2.40)$$

Полученная формула называется *вторым интерполяционным многочленом Ньютона для интерполирования назад*.

Рассмотрим пример применения интерполяционной формулы Ньютона при ручном счете.

Таблица 2.1

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
0	1.272	3.193	−2.014	1.000	−1.999	4.999
0.2	4.465	1.179	−1.014	−0.999	3.000	
0.4	5.644	0.165	−2.013	2.001		
0.6	5.809	−1.848	−0.012			
0.8	3.961	−1.860				
1.0	2.101					

Пример. Вычислить в точке $x = 0.1$ значение функции $y = f(x)$, заданной табл. 2.1.

Процесс вычислений удобно свести в ту же табл. 2.1. Каждая последующая конечная разность получается путем вычитания в предыдущей колонке верхней строки из нижней. При $x = 0.1$ имеем $t = (x - x_0)/h = (0.1 - 0)/0.2 = 0.5$. Проводя вычисления с пятью разрядами по формуле (2.39), получим

$$f(0.1) \approx N(0.1) = 1.272 + 0.5 \cdot 3.193 + \\ + \frac{0.5(0.5-1)}{2!} \cdot (-2.014) + \frac{0.5(0.5-1)(0.5-2)}{3!} \cdot 1.000 + \\ + \frac{0.5(0.5-1)(0.5-2)(0.5-3)}{4!} \cdot (-1.999) + \\ + \frac{0.5(0.5-1)(0.5-2)(0.5-3)(0.5-4)}{5!} \cdot 4.999 = \\ = 1.272 + 1.597 + 0.2518 + 0.06249 + 0.07806 + 0.1367 = 3.398.$$

Для сравнения проведем аналогичные вычисления по формуле (2.40). В этом случае $t = (x - x_n)/h = (0.1 - 1)/0.2 = -4.5$. Тогда

$$f(0.1) \approx N(0.1) = 2.101 - 4.5 \cdot (-1.860) + \\ + \frac{-4.5(-4.5+1)}{2!} \cdot (-0.012) + \frac{-4.5(-4.5+1)(-4.5+2)}{3!} \cdot 2.001 + \\ + \frac{-4.5(-4.5+1)(-4.5+2)(-4.5+3)}{4!} \cdot 3.000 + \\ + \frac{-4.5(-4.5+1)(-4.5+2)(-4.5+3)(-4.5+4)}{5!} \cdot 4.999 = \\ = 2.101 + 8.370 - 0.09450 - 13.13 + 7.383 - 1.231 = 3.402.$$

Видно, что здесь происходит заметная потеря точности. Если проводить вычисления более точно, то формулы (2.39) и (2.40) приведут к одному результату $f(0.1) \approx 3.3975$.

Мы рассмотрели построение интерполяционного многочлена Ньютона для равноотстоящих узлов. Можно построить многочлен Ньютона и для произвольно расположенных узлов, как и в случае многочлена Лагранжа. Однако этот случай мы рассматривать не будем.

В заключение отметим, что разные способы построения многочленов Лагранжа и Ньютона дают тождественные интерполяционные формулы при заданной таблице значений функции. Это следует из единственности интерполяционного многочлена заданной степени (при отсутствии совпадающих узлов интерполяции). Многочлен Ньютона удобно применять, например, если количество узлов интерполяции постепенно увеличивается. При этом учет нового узла требует лишь вычисления одного дополнительного слагаемого в (2.39) и (2.40), в то время как при использовании многочлена Лагранжа требуется пересчитывать все слагаемые в (2.34).

4. Точность интерполяции. График интерполяционного многочлена $y=\varphi(x)$ проходит через заданные точки, т. е. значения многочлена и данной функции $y=f(x)$ совпадают в узлах $x=x_i$ $(i=0,1,\ldots,n)$. Если функция $f(x)$ сама является многочленом степени n, то имеет место тождество $f(x)\equiv\varphi(x)$. В общем случае в точках, отличных от узлов интерполяции, $R(x)=f(x)-\varphi(x)\neq 0$. Эта разность есть погрешность интерполяции и называется *остаточным членом интерполяционной формулы*. Оценим его значение.

Предположим, что заданные числа y_i являются значениями некоторой функции $y=f(x)$ в точках $x=x_i$. Пусть эта функция непрерывна и имеет непрерывные производные до $(n+1)$ – го порядка включительно. Можно показать, что в этом случае остаточный член интерполяционного многочлена Лагранжа имеет вид

$$R_L(x)=\frac{(x-x_0)(x-x_1)\ldots(x-x_n)}{(n+1)!}\,f^{(n+1)}(x_*). \qquad (2.41)$$

Здесь $f^{(n+1)}(x_*)$ — производная $(n+1)$ – го порядка функции $f(x)$ в некоторой точке $x=x_*$, $x_*\in[x_0,x_n]$. Если максимальное значение этой производной равно

$$\max_{x_0\leqslant x\leqslant x_n}\left|f^{(n+1)}(x)\right|=M_{n+1},$$

то можно записать формулу для оценки остаточного члена:

$$|R_L(x)|\leqslant\frac{\omega_n(x)}{(n+1)!}\,M_{n+1},$$

где функция $\omega_n(x)$ определена как

$$\omega_n(x)=(x-x_0)(x-x_1)\ldots(x-x_n). \qquad (2.42)$$

Проанализировав поведение функции $\omega_n(x)$, можно сделать вывод о том, что погрешность интерполяции $R_L(x)$ в среднем будет тем выше, чем ближе точка x лежит к концам отрезка $[x_0,x_n]$. Если же использовать интерполяционный многочлен для аппроксимации функции вне отрезка $[x_0,x_n]$ (экстраполяция), то погрешность возрастет существенно.

Вид остаточного члена интерполяционного многочлена Ньютона в случае равноотстоящих узлов можно легко получить из (2.41):

$$R_N(x)=\frac{t(t-1)\ldots(t-n)}{(n+1)!}\,f^{(n+1)}(x_*)h^{n+1},\quad t=\frac{x-x_0}{h}\,.$$

Если предположить, что разность $\Delta^{n+1}y_n$ постоянна, то можно записать следующую формулу остаточного члена первого интерполяционного многочлена Ньютона:

$$R_N(x)=\frac{t(t-1)\ldots(t-n)}{(n+1)!}\,\Delta^{n+1}y_0.$$

Следует еще раз подчеркнуть, что существует один и только один интерполяционный многочлен при заданном наборе узлов интерполяции. Формулы Лагранжа, Ньютона и другие порождают один и тот же

многочлен (при условии, что вычисления проводятся точно). Разница лишь в алгоритме их построения. Правда, интерполяционный многочлен Лагранжа не содержит явных выражений для коэффициентов.

Выбор способа интерполяции определяется различными соображениями: точностью, временем вычислений, погрешностями округлений и др. В некоторых случаях более предпочтительной может оказаться локальная интерполяция, в то время как построение единого многочлена высокой степени (глобальная интерполяция) не приводит к успеху.

Такого рода ситуацию в 1901 г. обнаружил К. Рунге. Он строил на отрезке $-1 \leqslant x \leqslant 1$ интерполяционные многочлены с равномерным распределением узлов для функции $y = 1/(1 + 25x^2)$. Оказалось, что при увеличении степени интерполяционного многочлена последовательность его значений расходится для любой фиксированной точки x при $0.7 < x < 1$.

Положение в некоторых случаях может быть исправлено специальным распределением узлов интерполяции (если они не зафиксированы). Доказано, что если функция $f(x)$ имеет непрерывную производную на отрезке $[-1, 1]$, то при выборе значений x_i, совпадающих с корнями многочленов Чебышева степени $n + 1$, интерполяционные многочлены степени n сходятся к значениям функции в любой точке этого отрезка. Наглядно пояснить сделанное утверждение можно следующим образом. Как было отмечено в § 2, корни многочленов Чебышева расположены неравномерно на отрезке и сгущаются к его концам. Такое сгущение компенсирует увеличение погрешности интерполяции при приближении к концам отрезка, которое имеет место при равномерном распределении узлов.

Таким образом, повышение точности интерполяции целесообразно производить за счет уменьшения шага и специального расположения точек x_i. Повышение степени интерполяционного многочлена при локальной интерполяции также уменьшает погрешность, однако здесь не всегда ясно поведение производной $f^{(n+1)}(x)$ при увеличении n. Поэтому на практике стараются использовать многочлены малой степени (линейную и квадратичную интерполяции, сплайны).

5. Сплайны. Сейчас широкое распространение для интерполяции получило использование кубических *сплайн-функций* — специальным образом построенных многочленов третьей степени. Они представляют собой некоторую математическую модель гибкого тонкого стержня из упругого материала. Если закрепить его в двух соседних узлах интерполяции с заданными углами наклонов α и β (рис. 2.8), то между точками закрепления этот стержень (механический сплайн) примет некоторую форму, минимизирующую его потенциальную энергию.

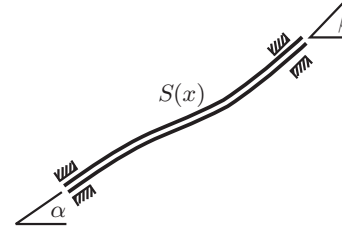


Рис. 2.8. Механический сплайн

Пусть форма этого стержня определяется функцией $y = S(x)$. Из курса сопротивления материалов известно, что уравнение свободного равновесия имеет вид $S^{\mathrm{IV}}(x)=0$. Отсюда следует, что между каждой парой соседних узлов интерполяции функция $S(x)$ является многочленом степени не выше третьей. Запишем ее в виде

$$S(x) = S_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + \\ + d_i(x - x_{i-1})^3, \quad x_{i-1} \leqslant x \leqslant x_i. \tag{2.43}$$

Для определения коэффициентов a_i, b_i, c_i, d_i на всех n элементарных отрезках необходимо получить $4n$ уравнений. Часть из них вытекает из условий прохождения графика функции $S(x)$ через заданные точки, т. е. $S_i(x_{i-1}) = y_{i-1}$, $S_i(x_i) = y_i$. Эти условия можно записать в виде

$$S_i(x_{i-1}) = a_i = y_{i-1}, \tag{2.44}$$

$$S_i(x_i) = a_i + b_i h + c_i h^2 + d_i h^3 = y_i, \tag{2.45}$$

$$h_i = x_i - x_{i-1}, \quad i = 1, 2, \ldots, n.$$

Эта система содержит $2n$ уравнений. Для получения недостающих уравнений зададим условия непрерывности первых и вторых производных во внутренних узлах интерполяции, т. е. условия гладкости второго порядка кривой во всех точках:

$$S_i'(x_i) = S_{i+1}'(x_i), \quad S_i''(x_i) = S_{i+1}''(x_i), \quad i = 1, 2, \ldots, n-1. \tag{2.46}$$

Вычислим производные многочлена (2.43):

$$S_i'(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2, \\ S_i''(x) = 2c_i + 6d_i(x - x_{i-1}). \tag{2.47}$$

Подставляя эти выражения в (2.46), получаем $2n - 2$ уравнений

$$b_{i+1} = b_i + 2h_i c_i + 3h_i^2 d_i, \tag{2.48}$$

$$c_{i+1} = c_i + 3h_i d_i, \tag{2.49}$$

$$i = 1, 2, \ldots, n-1.$$

Недостающие два уравнения получаются из условий закрепления концов сплайна. Обычно эти условия представляют собой соотношения, в которые входят значения первой и второй производных функции $S(x)$ в точках x_0 и x_n. Поэтому указанные значения должны входить в рассматриваемую систему уравнений. Из (2.47) следует, что $S_i'(x_{i-1}) = b_i$, $S_i''(x_{i-1}) = 2c_i$. Отсюда $S_1'(x_0) = b_1$, $S_1''(x_0) = 2c_1$, т. е. значения производных в точке x_0 присутствуют в системе. Значения же производных в точке x_n в системе отсутствуют. Введем их в систему с помощью дополнительных неизвестных b_{n+1} и c_{n+1}:

$$S'(x_n) = b_{n+1}, \quad S''(x_n) = 2c_{n+1}.$$

Из условий непрерывности производных в точке x_n следует, что

$$b_{n+1} = b_n + 2h_n c_n + 3h_n^2 d_n, \quad c_{n+1} = c_n + 3h_n d_n.$$

Таким образом, соотношения (2.48), (2.49) можно рассматривать для диапазона индексов

$$i = 1, 2, \ldots, n. \tag{2.50}$$

Система (2.44), (2.45), (2.48), (2.49) с учетом (2.50) содержит $4n+2$ неизвестных и $4n$ уравнений и может быть дополнена, например, следующими условиями закрепления концов сплайна:

$$S'(x_0) = b_1 = k_1, \quad S'(x_n) = b_{n+1} = k_2 \tag{2.51}$$

или

$$S''(x_0) = 2c_1 = m_1, \quad S''(x_n) = 2c_{n+1} = m_2, \tag{2.52}$$

где k_1, k_2, m_1, m_2 — заданные числа.

В частности, при заданных углах наклона α и β (см. рис. 2.8)

$$k_1 = \operatorname{tg}\alpha, \quad k_2 = \operatorname{tg}\beta.$$

При свободном закреплении концов можно приравнять нулю кривизну линии в точках закрепления. Получаемая таким образом функция называется *свободным кубическим сплайном*. Из условия нулевой кривизны на концах следуют равенства нулю вторых производных в этих точках. Отсюда

$$m_1 = m_2 = 0.$$

Соотношения (2.44), (2.45), (2.48), (2.49), а также (2.51) или (2.52) составляют систему линейных алгебраических уравнений для определения коэффициентов a_i, d_i $(i = 1, 2, \ldots, n)$ и b_i, c_i $(i = 1, 2, \ldots, n+1)$. Ее можно решить одним из методов, изложенных в гл. 4.

Однако с целью экономии памяти компьютера и машинного времени эту систему можно привести к более удобному виду. Из условия (2.44) сразу можно найти все коэффициенты a_i. Далее, из (2.49) получим

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad i = 1, 2, \ldots, n. \tag{2.53}$$

Подставим эти соотношения, а также значения $a_i = y_{i-1}$ в (2.45) и найдем отсюда коэффициенты

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i), \quad i = 1, 2, \ldots, n. \tag{2.54}$$

Учитывая выражения (2.53) и (2.54), исключаем из уравнения (2.48) коэффициенты d_i и b_i. Окончательно получим следующую систему уравнений только для коэффициентов c_i:

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right), \quad i = 2, 3, \ldots, n; \tag{2.55}$$

$$\frac{y_1 - y_0}{h_1} - \frac{h_1}{3}(c_2 + 2c_1) = k_1, \quad \frac{y_n - y_{n-1}}{h_n} + \frac{h_n}{3}(2c_{n+1} + c_n) = k_2 \tag{2.56}$$

или

$$c_1 = \frac{m_1}{2}, \quad c_{n+1} = \frac{m_2}{2}. \tag{2.57}$$

Здесь уравнения (2.56) используются при применении условий (2.51), а уравнения (2.57) — при применении условий (2.52).

Матрица системы (2.55) трехдиагональная, т. е. ненулевые элементы находятся лишь на главной и двух соседних с ней диагоналях, расположенных сверху и снизу. Для ее решения целесообразно использовать метод прогонки (см. гл. 4). По найденным из системы (2.55), (2.56) или (2.57) коэффициентам c_i легко вычислить коэффициенты d_i, b_i.

Заметим, что кубическая сплайн-функция, удовлетворяющая условиям (2.51) или (2.52), обладает наименьшей (в некотором смысле) кривизной среди всех дважды непрерывно дифференцируемых функций на отрезке $[x_0, x_n]$ с заданными значениями в узлах интерполяции, удовлетворяющих условиям (2.51) или (2.52). А именно,

$$\int_{x_0}^{x_n} [S''(x)]^2\, dx \leqslant \int_{x_0}^{x_n} [f''(x)]^2\, dx$$

для всех $f(x)$ из указанного класса функций.

6. О других формулах интерполяции. Ранее уже упоминалась одна из модификаций многочлена Лагранжа — интерполяционный многочлен Эрмита. При построении этого многочлена требуется, чтобы в узлах x_i совпадали с табличными данными не только его значения, но и значения его производных до некоторого порядка. В общем случае выражение для многочлена Эрмита очень громоздко, и пользоваться им на практике трудно. Поэтому ограничиваются лишь некоторыми простейшими случаями. Например, многочлен Эрмита, который сохраняет в двух точках ($x = x_0$, x_1) значения заданной функции $y = f(x)$ и ее первой производной $y' = f'(x)$, имеет вид

$$H(x) = y_0 + (x - x_0)\left\{y_0' + \frac{x - x_0}{x_0 - x_1}\left[\left(y_0 - \frac{y_0 - y_1}{x_0 - x_1}\right) + \right.\right.$$
$$\left.\left. + \frac{x - x_1}{x_0 - x_1}\left(y_0' - 2\frac{y_0 - y_1}{x_0 - x_1} + y_1'\right)\right]\right\}.$$

Иногда при выводе интерполяционных формул удобнее использовать не односторонние разности, как для многочлена Ньютона, а центральные. На этом основаны интерполяционные формулы Стирлинга и Бесселя. Они могут быть получены путем преобразования формулы Ньютона.

Рассмотрим интерполирование функций специального вида, а именно *периодических функций*. Для функции с периодом 2π можно построить

интерполяционную формулу по аналогии с формулой Лагранжа:

$$F(x)=\frac{\sin(x-x_1)\sin(x-x_2)\dots\sin(x-x_n)}{\sin(x_0-x_1)\sin(x_0-x_2)\dots\sin(x_0-x_n)}\,y_0+$$
$$+\frac{\sin(x-x_0)\sin(x-x_2)\dots\sin(x-x_n)}{\sin(x_1-x_0)\sin(x_1-x_2)\dots\sin(x_1-x_n)}\,y_1+\dots$$
$$\dots+\frac{\sin(x-x_0)\sin(x-x_1)\dots\sin(x-x_{n-1})}{\sin(x_n-x_0)\sin(x_n-x_1)\dots\sin(x_n-x_{n-1})}\,y_n.$$

7. Функции двух переменных. До сих пор мы рассматривали интерполирование функций одной независимой переменной $y=f(x)$. На практике возникает также необходимость построения интерполяционных формул для функций нескольких переменных. Для простоты ограничимся функцией двух переменных $z=f(x,y)$. Пусть ее значения заданы на множестве равноотстоящих узлов (x_i,y_i) $(i,j=0,1,2)$. Введем обозначения $z_{ij}=f(x_i,y_j)$, $h_1=x_{i+1}-x_i$, $h_2=y_{j+1}-y_j$.

Построим многочлен, аналогичный многочлену Ньютона для случая одной переменной. Здесь нужно вычислять разности двух видов — по направлениям x и y. Эти *частные* разности первого порядка определяются формулами

$$\Delta_x z_{ij}=z_{i+1,j}-z_{ij},\quad \Delta_y z_{ij}=z_{i,j+1}-z_{ij}.$$

Запишем также выражения для частных разностей второго порядка:

$$\Delta^2_{xx}z_{ij}=z_{i+2,j}-2z_{i+1,j}+z_{ij},$$
$$\Delta^2_{yy}z_{ij}=z_{i,j+2}-2z_{i,j+1}+z_{ij},$$
$$\Delta^2_{xy}z_{ij}=(z_{i+1,j+1}-z_{i,j+1})-(z_{i+1,j}-z_{ij}).$$

Интерполяционный многочлен второй степени можно записать в виде

$$F(x,y)=z_{00}+\frac{x-x_0}{h_1}\,\Delta_x z_{00}+\frac{y-y_0}{h_2}\,\Delta_y z_{00}+$$
$$+\frac{(x-x_0)(x-x_1)}{2h_1^2}\,\Delta^2_{xx}z_{00}+\frac{(x-x_0)(y-y_0)}{h_1h_2}\,\Delta^2_{xy}z_{00}+$$
$$+\frac{(y-y_0)(y-y_1)}{2h_2^2}\,\Delta^2_{yy}z_{00}.$$

Можно также построить *линейную интерполяционную формулу*. Геометрически это означает, что нужно найти уравнение плоскости, проходящей через три заданные точки (x_i,y_i,z_i), $(i=1,\ 2,\ 3)$, где $z_i=f(x_i,y_i)$. Из курса аналитической геометрии известно, что уравнение плоскости, проходящей через три точки, можно записать в виде

$$\begin{vmatrix} x-x_1 & y-y_1 & z-z_1\\ x_2-x_1 & y_2-y_1 & z_2-z_1\\ x_3-x_1 & y_3-y_1 & z_3-z_1\end{vmatrix}=0.$$

Отсюда можно найти

$$z = \frac{1}{D_3}(D_0 - D_1 x - D_2 y), \tag{2.58}$$

$$D_0 = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}, \quad D_1 = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix},$$

$$D_2 = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix}, \quad D_3 = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}.$$

Пример. Вычислить приближенное значение функции $z = f(x, y)$ в точке $(1, 0)$, если известны ее значения $z_1 = f(0, 0) = 0$, $z_2 = f(2, 4) = -3$, $z_3 = f(4, -2) = 1$.

Воспользуемся формулой линейной интерполяции (2.58). Вычислим значения определителей

$$D_0 = \begin{vmatrix} 0 & 0 & 0 \\ 2 & 4 & -3 \\ 4 & -2 & 1 \end{vmatrix} = 0, \quad D_1 = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 4 & -3 \\ 1 & -2 & 1 \end{vmatrix} = -2,$$

$$D_2 = \begin{vmatrix} 0 & 1 & 0 \\ 2 & 1 & -3 \\ 4 & 1 & 1 \end{vmatrix} = -14, \quad D_3 = \begin{vmatrix} 0 & 0 & 1 \\ 2 & 4 & 1 \\ 4 & -2 & 1 \end{vmatrix} = -20.$$

Таким образом, $z \approx -(2x + 14y)/20$, или $z \approx -0.1x - 0.7y$. Это и есть формула линейной интерполяции для данного примера. При $x = 1$, $y = 0$ получим $z \approx -0.1$.

§ 4. Подбор эмпирических формул

1. Характер опытных данных. При интерполировании функций мы использовали условие равенства значений интерполяционного многочлена и данной функции в известных точках — узлах интерполяции. Это предъявляет высокие требования к точности данных значений функции. В случае обработки опытных данных, полученных в результате наблюдений или измерений, нужно иметь в виду *ошибки* этих данных. Они могут быть вызваны несовершенством измерительного прибора, субъективными причинами, различными случайными факторами и т. д. Ошибки экспериментальных данных можно условно разбить на три категории по их происхождению и величине: систематические, случайные и грубые.

Систематические ошибки обычно дают отклонение в одну сторону от истинного значения измеряемой величины. Они могут быть постоянными или закономерно изменяться при повторении опыта, и их причина и характер известны. Систематические ошибки могут быть вызваны условиями эксперимента (влажностью, температурой среды и др.), дефектом

измерительного прибора, его плохой регулировкой (например, смещением указательной стрелки от нулевого положения) и т. д. Эти ошибки можно устранить наладкой аппаратуры или введением соответствующих поправок.

Случайные ошибки определяются большим числом факторов, которые не могут быть устранены либо достаточно точно учтены при измерениях или при обработке результатов. Они имеют случайный (несистематический) характер, дают отклонения от средней величины в ту и другую стороны при повторении измерений и не могут быть устранены в эксперименте, как бы тщательно он ни проводился. С вероятностной точки зрения математическое ожидание случайной ошибки равно нулю. Статистическая обработка экспериментальных данных позволяет определить величину случайной ошибки и довести ее до некоторого приемлемого значения повторением измерений достаточное число раз.

Грубые ошибки явно искажают результат измерения; они чрезмерно большие и обычно пропадают при повторении опыта. Грубые ошибки существенно выходят за пределы случайной ошибки, полученные при статистической обработке. Измерения с такими ошибками отбрасываются и в расчет при окончательной обработке результатов измерений не принимаются.

Таким образом, в экспериментальных данных всегда имеются случайные ошибки. Они, вообще говоря, могут быть уменьшены до сколь угодно малой величины путем многократного повторения опыта. Однако это не всегда целесообразно, поскольку могут потребоваться большие материальные или временны́е ресурсы. Значительно дешевле и быстрее можно в ряде случаев получить уточненные данные хорошей математической обработкой имеющихся результатов измерений.

В частности, с помощью статистической обработки результатов измерений можно найти закон распределения ошибок измерений, наиболее вероятный диапазон изменения искомой величины (доверительный интервал) и другие параметры. Рассмотрение этих вопросов выходит за рамки данного пособия; их изложение можно найти в некоторых книгах, приведенных в списке литературы. Здесь ограничимся лишь определением связи между исходным параметром x и искомой величиной y на основании результатов измерений.

2. Эмпирические формулы. Пусть, изучая неизвестную функциональную зависимость между y и x, мы в результате серии экспериментов произвели ряд измерений этих величин и получили таблицу значений

x_0	x_1	$\dots$	x_n
y_0	y_1	$\dots$	y_n

Задача состоит в том, чтобы найти приближенную зависимость

$$y = f(x), \tag{2.59}$$

значения которой при $x = x_i$ $(i = 0, 1, \ldots, n)$ мало отличаются от опытных данных y_i. Приближенная функциональная зависимость (2.59), полученная на основании экспериментальных данных, называется *эмпирической формулой*.

В § 1 отмечалось, что задача построения эмпирической формулы отличается от задачи интерполирования. График эмпирической зависимости, вообще говоря, не проходит через заданные точки (x_i, y_i), как в случае интерполяции. Это приводит к тому, что экспериментальные данные в некоторой степени сглаживаются, в то время как интерполяционная формула повторила бы все ошибки, имеющиеся в экспериментальных данных.

Построение эмпирической формулы состоит из двух этапов: подбора общего вида этой формулы и определения наилучших значений содержащихся в ней параметров. Общий вид формулы иногда известен из физических соображений. Например, для упругой среды связь между напряжением σ и относительной деформацией ε определяется законом Гука: $\sigma = E\varepsilon$, где E — модуль упругости; задача сводится к определению одного неизвестного параметра E.

Если характер зависимости неизвестен, то вид эмпирической формулы может быть произвольным. Предпочтение обычно отдается наиболее простым формулам, обладающим достаточной точностью. Они первоначально выбираются из геометрических соображений: экспериментальные точки наносятся на график, и примерно угадывается общий вид зависимости путем сравнения полученной кривой с графиками известных функций (многочлена, показательной или логарифмической функций и т. п.).

Успех здесь в значительной мере определяется опытом и интуицией исследователя.

Простейшей эмпирической формулой является линейная зависимость

$$y = ax + b. \tag{2.60}$$

Близость экспериментального распределения точек к линейной зависимости легко просматривается после построения графика данной экспериментальной зависимости. Кроме того, эту зависимость можно проверить путем вычисления значений k_i:

$$k_i = \Delta y_i / \Delta x_i, \quad \Delta y_i = y_{i+1} - y_i, \quad \Delta x_i = x_{i+1} - x_i,$$
$$i = 0, 1, \ldots, n-1.$$

Если при этом $k_i \approx \text{const}$, то точки (x_i, y_i) расположены приблизительно на одной прямой, и может быть поставлен вопрос о применимости эмпирической формулы (2.60). Точность такой аппроксимации определяется отклонением величин k_i от постоянного значения. В частном случае равноотстоящих точек x_i (т. е. $\Delta x_i = \text{const}$) достаточно проверить постоянство разностей Δy_i.

Пример. Проверим возможность использования линейной зависимости для описания следующих данных:

x	0	0.5	1.0	1.5	2.0	2.5
y	1.17	1.81	2.50	3.15	3.79	4.44

Поскольку здесь x_i — равноотстоящие точки ($\Delta x_i = x_{i+1} - x_i = 0.5$), то достаточно вычислить разности Δy_i: 0.64, 0.69, 0.65, 0.64, 0.65. Так как эти значения близки друг к другу, то в качестве эмпирической формулы можно принять линейную зависимость.

В ряде случаев к линейной зависимости могут быть сведены и другие экспериментальные данные, когда их график в декартовой системе координат не является прямой линией. Это может быть достигнуто путем *введения новых переменных* ξ, η вместо x, y:

$$\xi = \varphi(x, y), \quad \eta = \psi(x, y). \tag{2.61}$$

Функции $\varphi(x, y)$ и $\eta = \psi(x, y)$ выбираются такими, чтобы точки (ξ_i, η_i) лежали на некоторой прямой линии в плоскости (ξ, η). Такое преобразование называется *выравниванием данных*.

Для получения линейной зависимости

$$\eta = a\xi + b$$

с помощью преобразования (2.61) исходная формула должна быть записана в виде

$$\psi(x, y) = a\varphi(x, y) + b.$$

К такому виду легко сводится, например, степенная зависимость $y = ax^b$, ($x > 0$, $y > 0$). Логарифмируя эту формулу, получаем $\lg y = b \lg x + \lg a$. Полагая $\xi = \lg x$, $\eta = \lg y$, находим линейную связь: $\eta = b\xi + c$ ($c = \lg a$).

Другой простейшей эмпирической формулой является квадратный трехчлен

$$y = ax^2 + bx + c. \tag{2.62}$$

Возможность использования этой формулы для случая равноотстоящих точек x_i можно проверить путем вычисления вторых разностей $\Delta^2 y_i = y_{i+2} - 2y_{i+1} + y_i$. При $\Delta^2 y_i \approx \text{const}$ в качестве эмпирической формулы может быть выбрана (2.62).

3. Определение параметров эмпирической зависимости. Будем считать, что тип эмпирической формулы выбран, и ее можно представить в виде

$$y = \varphi(x, a_0, a_1, \ldots, a_m), \tag{2.63}$$

где φ — известная функция, $a_0, a_1, \ldots, a_m$ — неизвестные постоянные параметры. Задача состоит в том, чтобы определить такие значения этих параметров, при которых эмпирическая формула дает хорошее приближение данной функции, значения которой в точках x_i равны y_i ($i = 0, 1, \ldots, n$).

Как уже отмечалось выше, здесь не ставится условие (как в случае интерполяции) совпадения опытных данных y_i со значениями эмпирической функции (2.63) в точках x_i. Разность между этими значениями (отклонения) обозначим через ε_i. Тогда

$$\varepsilon_i = \varphi(x_i, a_0, a_1, \ldots, a_m) - y_i, \quad i = 0, 1, \ldots, n. \tag{2.64}$$

Задача нахождения наилучших значений параметров $a_0, a_1, \ldots, a_m$ сводится к некоторой минимизации отклонений ε_i. Существует несколько способов решения этой задачи.

Простейшим из них является *метод выбранных точек*. Он состоит в следующем. По заданным экспериментальным значениям на координатной плоскости OXY наносится система точек. Затем проводится простейшая плавная линия (например, прямая), которая наиболее близко примыкает к данным точкам. На этой линии выбираются точки, которые, вообще говоря, не принадлежат исходной системе точек. Число выбранных точек должно быть равным количеству искомых параметров эмпирической зависимости. Координаты этих точек (x_j^0, y_j^0) тщательно измеряются и используются для записи условия прохождения графика эмпирической функции (2.63) через выбранные точки:

$$\varphi(x_j^0, a_0, a_1, \ldots, a_m) = y_j^0, \quad j = 0, 1, \ldots, m. \tag{2.65}$$

Из этой системы уравнений находятся значения параметров a_0, $a_1, \ldots, a_m$.

В частности, если в качестве эмпирической формулы принята линейная зависимость $y = ax + b$, то на этой прямой выбираются точки (x_0^0, y_0^0) и (x_1^0, y_1^0), и уравнения (2.65) примут вид

$$\begin{aligned} ax_0^0 + b &= y_0^0, \\ ax_1^0 + b &= y_1^0. \end{aligned} \tag{2.66}$$

Можно также сразу записать уравнение прямой, проходящей через эти выбранные точки. В этом случае не нужно решать систему (2.66)

Рассмотрим еще один способ определения параметров эмпирической формулы — *метод средних*. Он состоит в том, что параметры a_0, $a_1, \ldots, a_m$ зависимости (2.63) определяются с использованием условия равенства нулю суммы отклонений (2.64) во всех точках x_i:

$$\sum_{i=0}^{n} \varepsilon_i = \sum_{i=0}^{n} [\varphi(x_i, a_0, a_1, \ldots, a_m) - y_i] = 0. \tag{2.67}$$

Полученное уравнение служит для определения параметров $a_0, a_1, \ldots, a_m$. Ясно, что из одного уравнения нельзя однозначно определить все $m + 1$ параметров. Однако, поскольку других условий нет, равенство (2.67) путем группировки отклонений ε_i, разбивается на систему, состоящую из $m + 1$ уравнений. Например,

$$\varepsilon_0 + \varepsilon_1 + \varepsilon_2 = 0,$$
$$\varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6 = 0,$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$\varepsilon_{n-1} + \varepsilon_n = 0.$$

Решая эту систему уравнений, можно найти неизвестные параметры.

Пример. Тело, движущееся прямолинейно с неизвестной скоростью v_0, в момент времени $t = 0$ начинает тормозить. Измерялось расстояние x от начала торможения в следующие моменты времени t:

t, с	0	5	10	15	20	25
x, м	0	106	182	234	261	275

Считая движение тела равнозамедленным с постоянным замедлением $-a$ (т. е. с ускорением a), найти приближенные значения параметров v_0 и a.

Решение. Искомые параметры могут быть найдены из уравнения движения тела, которое представим с помощью эмпирической формулы, используя результаты измерений. Вид эмпирической формулы в данном случае известен из физических соображений — при равнозамедленном движении тела пройденное расстояние является квадратичной функцией времени:

$$x = At^2 + Bt + C.$$

Легко установить, что $C = 0$, поскольку $x = 0$ при $t = 0$. Эмпирическая формула принимает вид

$$x = At^2 + Bt. \tag{2.68}$$

Для определения параметров A, B нужно получить два уравнения. Воспользуемся методом средних и запишем уравнение (2.67) для всех точек (кроме начальной):

$$\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + \varepsilon_5 = 0.$$

Запишем вместо этого уравнения систему двух уравнений путем его расщепления:

$$\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 0,$$
$$\varepsilon_4 + \varepsilon_5 = 0.$$

Используя выражение (2.68) и табличные данные, получаем

$$(A \cdot 5^2 + B \cdot 5 - 106) + (A \cdot 10^2 + B \cdot 10 - 182) + (A \cdot 15^2 + B \cdot 15 - 234) = 0,$$
$$(A \cdot 20^2 + B \cdot 20 - 261) + (A \cdot 25 + B \cdot 25 - 275) = 0.$$

Или окончательно

$$175A + 15B = 261,$$
$$1025A + 45B = 536.$$

Решая эту систему уравнений, находим $A = -0.30$, $B = 39.07$.

Следовательно, эмпирическую формулу (2.68), которая дает приближенную связь между пройденным расстоянием и временем, можно записать в виде

$$x = -0.30t^2 + 39.07t.$$

Сравнивая это уравнение с уравнением $x = at^2/2 + v_0 t$, получаем оценки для среднего ускорения тела и его начальной скорости:

$$a = 2A = -0.60 \text{ м/с}^2, \quad v_0 = B = 39.07 \text{ м/с}.$$

Рассмотренные методы определения параметров эмпирической формулы являются сравнительно простыми, однако в ряде случаев получаемые с их помощью аппроксимации не обладают достаточной точностью.

4. Метод наименьших квадратов. Запишем сумму квадратов отклонений (2.64) для всех точек $x_0, x_1, \ldots, x_n$:

$$S = \sum_{i=0}^{n} \varepsilon_i^2 = \sum_{i=0}^{n} [\varphi(x_i, a_0, a_1, \ldots, a_m) - y_i]^2. \tag{2.69}$$

Параметры $a_0, a_1, \ldots, a_m$ эмпирической формулы (2.63) будем находить из условия минимума функции $S = S(a_0, a_1, \ldots, a_m)$. В этом состоит *метод наименьших квадратов*.

В теории вероятностей доказывается, что полученные таким методом значения параметров наиболее вероятны, если отклонения ε_i подчиняются нормальному закону распределения.

Поскольку здесь параметры $a_0, a_1, \ldots, a_m$ выступают в роли независимых переменных функции S, то ее минимум найдем, приравнивая нулю частные производные по этим переменным:

$$\frac{\partial S}{\partial a_0} = 0, \quad \frac{\partial S}{\partial a_1} = 0, \quad \ldots, \quad \frac{\partial S}{\partial a_m} = 0. \tag{2.70}$$

Полученные соотношения — система уравнений для определения $a_0, a_1, \ldots, a_m$.

Рассмотрим применение метода наименьших квадратов для широко используемого на практике частного случая, когда функция φ является линейной по неизвестным параметрам a_0, a_1, $\ldots$, a_m:

$$\varphi(x, a_0, a_1, \ldots, a_m) = \sum_{j=0}^{m} a_j \varphi_j(x),$$

где $\varphi_0, \varphi_1, \ldots, \varphi_m$ — известные функции x. Формула (2.69) для определения суммы квадратов отклонений S примет вид

$$S = \sum_{i=0}^{n} \Big[\sum_{j=0}^{m} a_j \varphi_j(x_i) - y_i\Big]^2.$$

Для составления системы (2.70) продифференцируем S по переменным a_k ($k = 0, 1, \ldots, m$):

$$\frac{\partial S}{\partial a_k} = \frac{\partial}{\partial a_k} \sum_{i=0}^{n} \Big[\sum_{j=0}^{m} a_j \varphi_j(x_i) - y_i\Big]^2 = \sum_{i=0}^{n} \frac{\partial}{\partial a_k} \Big[\sum_{j=0}^{m} a_j \varphi_j(x_i) - y_i\Big]^2 =$$

$$= \sum_{i=0}^{n} 2\Big[\sum_{j=0}^{m} a_j \varphi_j(x_i) - y_i\Big] \varphi_k(x_i) = 2 \sum_{i=0}^{n} \varphi_k(x_i) \Big[\sum_{j=0}^{m} a_j \varphi_j(x_i) - y_i\Big].$$

Приравнивая найденные производные нулю, получим следующую систему уравнений:

$$\sum_{i=0}^{n} \varphi_k(x_i) \Big[\sum_{j=0}^{m} a_j \varphi_j(x_i) - y_i\Big] = 0, \quad k = 0, 1, \ldots, m. \tag{2.71}$$

Система (2.71) является системой линейных алгебраических уравнений, ее можно записать в наглядном векторно-матричном виде (см. гл. 4). Для этого введем векторы опытных данных $\mathbf{y}$ и неизвестных параметров $\mathbf{a}$, а также матрицу Φ следующим образом

$$\mathbf{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}, \quad \Phi = \begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \ldots & \varphi_m(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \ldots & \varphi_m(x_1) \\ \ldots & \ldots & \ldots & \ldots \\ \varphi_0(x_n) & \varphi_1(x_n) & \ldots & \varphi_m(x_n) \end{pmatrix}.$$

Здесь векторы $\mathbf{y}$ и $\mathbf{a}$ имеют размерности $n+1$ и $m+1$ соответственно, а матрица Φ имеет размерность $(n+1) \times (m+1)$. Для ее элементов справедливо выражение

$$\Phi_{ij} = \varphi_j(x_i).$$

Нетрудно убедиться, что выражение, стоящее в квадратных скобках в (2.71), является i-й компонентой вектора $\Phi\mathbf{a} - \mathbf{y}$, а каждое уравнение (2.71) представляет собой равенство нулю k-й компоненты вектора $\Phi^{\mathrm{T}}(\Phi\mathbf{a} - \mathbf{y})$, где Φ^{T} — транспонированная матрица. Таким образом, систему (2.71) можно записать в виде

$$\Phi^{\mathrm{T}}(\Phi\mathbf{a} - \mathbf{y}) = 0$$

или

$$(\Phi^{\mathrm{T}}\Phi)\,\mathbf{a} = \Phi^{\mathrm{T}}\mathbf{y}. \tag{2.72}$$

Матрица этой системы $\Phi^{\mathrm{T}}\Phi$ имеет размерность $(m+1) \times (m+1)$, вектор $\mathbf{a}$ является искомым.

Пример. Используя метод наименьших квадратов, вывести эмпирическую формулу для функции $y = f(x)$, заданной в табличном виде:

x	0.75	1.50	2.25	3.00	3.75
y	2.50	1.20	1.12	2.25	4.28

Р е ш е н и е. Если изобразить данные табличные значения на графике (рис. 2.9), то легко убедиться, что в качестве эмпирической формулы для аппроксимации функции $y=f(x)$ можно принять квадратный трехчлен, графиком которого является парабола:

$$y \approx \varphi(x) = a_0 + a_1 x + a_2 x^2.$$

В данном случае имеем

$$m = 2, \quad n = 4, \quad \varphi_0(x) = 1, \quad \varphi_1(x) = x, \quad \varphi_2(x) = x^2,$$

$$\mathbf{y} = \begin{pmatrix} 2.50 \\ 1.20 \\ 1.12 \\ 2.25 \\ 4.28 \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \quad \Phi = \begin{pmatrix} 1 & 0.75 & 0.75^2 \\ 1 & 1.50 & 1.50^2 \\ 1 & 2.25 & 2.25^2 \\ 1 & 3.00 & 3.00^2 \\ 1 & 3.75 & 3.75^2 \end{pmatrix}.$$

После вычисления матрицы $\Phi^{\mathrm{T}}\Phi$ и вектора $\Phi^{\mathrm{T}}\mathbf{y}$ (приведены округленные значения)

$$\Phi^{\mathrm{T}}\Phi = \begin{pmatrix} 5 & 11.25 & 30.94 \\ 11.25 & 30.94 & 94.92 \\ 30.94 & 94.92 & 309.76 \end{pmatrix}, \quad \Phi^{\mathrm{T}}\mathbf{y} = \begin{pmatrix} 11.35 \\ 29.00 \\ 90.21 \end{pmatrix}$$

система уравнений (2.72) принимает вид

$$\begin{aligned} 5a_0 + 11.25a_1 + 30.94a_2 &= 11.35, \\ 11.25a_0 + 30.94a_1 + 94.92a_2 &= 29.00, \\ 30.94a_0 + 94.92a_1 + 309.76a_2 &= 90.21. \end{aligned}$$

Отсюда находим значения параметров эмпирической формулы: $a_0 = 4.82$, $a_1 = -3.88$, $a_2 = 1.00$. Таким образом, получаем следующую аппроксимацию функции, заданной в табличном виде:

$$y \approx 4.82 - 3.88x + 1.00x^2.$$

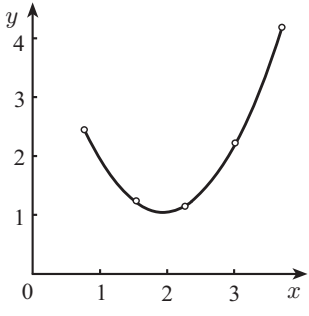


Рис. 2.9

Оценим относительные погрешности полученной аппроксимации в заданных точках, т. е. найдем значения

$$\delta y_i = \frac{\varepsilon_i}{y_i} = \frac{\varphi(x_i)}{y_i}.$$

Результаты вычислений представим в виде таблицы

x	$\varphi(x)$	y	ε	δy
0.75	2.47	2.50	−0.03	−0.012
1.50	1.25	1.20	0.05	0.042
2.25	1.15	1.12	0.03	0.027
3.00	2.17	2.25	−0.08	−0.036
3.75	4.32	4.28	0.04	0.009

На рис. 2.9 построен график найденной эмпирической функции. Точками, как уже отмечалось, нанесены заданные табличные значения функции $y = f(x)$.

В заключение сделаем несколько замечаний, касающихся метода наименьших квадратов.

Замечание 1. Можно доказать, что если столбцы матрицы Φ линейно независимы, то система (2.72) имеет единственное решение.

Замечание 2. Как видно из рассмотренного примера, матрица $\Phi^{\mathrm{T}}\Phi$ является *симметрической*, т. е. $(\Phi^{\mathrm{T}}\Phi)_{ij} = (\Phi^{\mathrm{T}}\Phi)_{ji}$ $(i, j = 0, 1, \ldots, m)$.

Замечание 3. В случае использования в качестве эмпирической функции многочлена

$$\varphi(x) = a_0 + a_1 x + \ldots + a_m x^m$$

элементы матрицы $\Phi^{\mathrm{T}}\Phi$ и компоненты вектора $\Phi^{\mathrm{T}}\mathbf{y}$ можно вычислить по формулам

$$(\Phi^{\mathrm{T}}\Phi)_{ij} = \sum_{k=0}^{n} x_k^{i+j}, \quad (\Phi^{\mathrm{T}}\mathbf{y})_i = \sum_{k=0}^{n} x_k^i y_k, \quad i, j = 0, 1, \ldots, m. \qquad (2.73)$$

В частности, равны все элементы $(\Phi^{\mathrm{T}}\Phi)_{ij}$ при $i + j = \mathrm{const}$.

5. Локальное сглаживание данных. Как отмечалось в п. 1, опытные данные содержат случайные ошибки, что является причиной разброса этих данных. Во многих случаях бывает целесообразно провести их *сглаживание* для получения более плавного характера исследуемой зависимости. Существуют различные способы сглаживания. Рассмотрим один из них, основанный на методе наименьших квадратов.

Пусть в результате экспериментального исследования зависимости $y = f(x)$ получена таблица значений искомой функции y_0, y_1, … … , y_n в точках x_0, x_1, … , x_n. Значения аргумента x_i предполагаются равноотстоящими, а опытные данные y_i — имеющими одинаковую точность. Предполагается также, что функция $y = f(x)$ на произвольной части отрезка $[x_0, x_n]$ может быть достаточно хорошо аппроксимирована многочленом некоторой степени m.

Рассматриваемый способ сглаживания состоит в следующем. Для нахождения сглаженного значения $\bar{y}_i$ в точке x_i выбираем по обе стороны от нее $k+1$ значение аргумента из имеющихся в таблице (k четное):

$x_{i-k/2}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{i+k/2}$. По опытным значениям рассматриваемой функции в этих точках $y_{i-k/2}, \ldots, y_{i-1}, y_i, y_{i+1}, \ldots, y_{i+k/2}$ строим многочлен степени m с помощью метода наименьших квадратов (при этом $m \leqslant k$). Значение полученного многочлена $\bar{y}_i$ в точке x_i и будет искомым (сглаженным) значением. Процесс повторяется для всех внутренних точек. Сглаживание значений, расположенных вблизи концов отрезка $[x_0, x_n]$, производится с помощью крайних точек.

Опыт показывает, что сглаженные значения $\bar{y}_i$, как правило, с достаточной степенью точности близки к истинным значениям. Иногда сглаживание повторяют. Однако это может привести к существенному искажению истинного характера рассматриваемой функциональной зависимости.

Приведем в заключение несколько формул для вычисления сглаженных значений опытных данных при различных m, k:

$$m = 1:$$

$$\bar{y}_i = \frac{1}{3}(y_{i-1} + y_i + y_{i+1}), \quad k = 2,$$

$$\bar{y}_i = \frac{1}{5}(y_{i-2} + y_{i-1} + y_i + y_{i+1} + y_{i+2}), \quad k = 4,$$

$$\bar{y}_i = \frac{1}{7}(y_{i-3} + y_{i-2} + y_{i-1} + y_i + y_{i+1} + y_{i+2} + y_{i+3}), \quad k = 6;$$

$$m = 3:$$

$$\bar{y}_i = \frac{1}{35}(-3y_{i-2} + 12y_{i-1} + 17y_i + 12y_{i+1} - 3y_{i+2}), \quad k = 4,$$

$$\bar{y}_i = \frac{1}{21}(-2y_{i-3} + 3y_{i-2} + 6y_{i-1} + 7y_i + 6y_{i+1} + 3y_{i+2} - 2y_{i+3}), \quad k = 6,$$

$$\bar{y}_i = \frac{1}{231}(-21y_{i-4} + 14y_{i-3} + 39y_{i-2} + 54y_{i-1} + 59y_i + 54y_{i+1} + 39y_{i+2} + 14y_{i+3} - 21y_{i+4}), \quad k = 8;$$

$$m = 5:$$

$$\bar{y}_i = \frac{1}{231}(5y_{i-3} - 30y_{i-2} + 75y_{i-1} + 131y_i + 75y_{i+1} - 30y_{i+2} + 5y_{i+3}), \quad k = 6,$$

$$\bar{y}_i = \frac{1}{429}(15y_{i-4} - 55y_{i-3} + 30y_{i-2} + 135y_{i-1} + 179y_i + 135y_{i+1} + 30y_{i+2} - 55y_{i+3} + 15y_{i+4}), \quad k = 8.$$

Упражнения

1. Записать алгоритмы вычислений с помощью разложений в ряды значений функций: а) $y = \cos x$; б) $y = e^{-x}$; в) $y = \operatorname{sh} x$; г) $y = \sqrt{1+x}$.

2. Преобразовать приближенно данные многочлены в многочлены третьей степени: а) $P(x) = x^5 - 3x^4 + 4$; б) $P(x) = x^4 + 5x^3 - 1$. Оценить допущенные погрешности.

3. Записать по схеме Горнера алгоритм вычисления первых пяти членов степенных рядов при разложении функций: а) $y = \sin x$; б) $y = \operatorname{ch} x$.

4. Используя цепные дроби, вычислить значения: а) $\ln 2$; б) $\operatorname{tg}(\pi/8)$; в) $e^{0.1}$; г) $\operatorname{arctg} 0.5$.

5. Дана таблица значений функции

x	0	0.2	0.4	0.6
y	1.763	1.917	2.143	2.362

а) С помощью линейной и квадратичной интерполяций найти приближенное значение функции при $x = 0.25$.

б) Вычислить, при каком значении аргумента справедливо равенство $y = 2.000$.

6. Убедиться, что формула линейной интерполяции (2.35) полученная как частный случай многочлена Лагранжа, эквивалентна выведенной ранее формуле (2.28).

7. Записать алгоритм вычисления функции с помощью квадратичной интерполяции. Алгоритм должен работать при любом значении аргумента x.

8. Построить интерполяционный многочлен Лагранжа для функции, заданной таблицей в упр. 5.

9. Вычислить значение функции, заданной в упр. 5, при $x = 0.1$, используя интерполяционный многочлен Ньютона. Оценить погрешность результата.

10. Построив график функции $\omega_n(x)$, заданной формулой (2.42), проверить, как изменяется погрешность интерполяции при изменении x.

11. Найти величину ускорения при равноускоренном движении тела, если известны значения пройденного им пути S в некоторые моменты времени t:

t, с	0	5	10	15	20	25
S, м	5	150	560	1200	2100	3300

12. Закон Гука можно записать в виде $\sigma = E\varepsilon$, где σ — напряжение, E — модуль упругости, ε — относительная деформация. При испытаниях образца произвели n измерений значений σ и ε. Написать алгоритм вычисления параметра E.

13. Изучается зависимость между, электродвижущей силой E и температурой нагревания T термопары. Данные измерений приведены в следующей таблице:

T, °C	500	750	1000	1250	1500
E, мВ	3.23	4.52	5.71	10.17	18.49

Найти приближенную зависимость $E(T)$ в виде квадратного трехчлена.

14*. Получить соотношения (2.73).

ГЛАВА 3

ДИФФЕРЕНЦИРОВАНИЕ И ИНТЕГРИРОВАНИЕ

§ 1. Численное дифференцирование

1. Аппроксимация производных. Напомним, что *производной* функции $y = f(x)$ называется предел отношения приращения функции Δy к приращению аргумента Δx при стремлении Δx к нулю:

$$y' = f'(x) = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}, \quad \Delta y = f(x + \Delta x) - f(x). \tag{3.1}$$

Обычно для вычисления производных используют готовые формулы (таблицу производных) и к выражению (3.1) не прибегают. Однако в численных расчетах на компьютере использование этих формул не всегда удобно и возможно. В частности, функция $y = f(x)$ может быть задана в виде таблицы значений (полученных, например, в результате численного расчета). В таких случаях производную можно найти, опираясь на формулу (3.1). Значение шага Δx полагают равным некоторому конечному числу, и для вычисления значения производной получают приближенное равенство

$$y' \approx \Delta y / \Delta x. \tag{3.2}$$

Это соотношение называется *аппроксимацией* (*приближением*) *производной с помощью отношения конечных разностей* (значения Δy, Δx в формуле (3.2) конечные в отличие от их бесконечно малых значений в (3.1)).

Рассмотрим аппроксимацию производной для функции $y = f(x)$, заданной в табличном виде: $y = y_0$, y_1, ..., в узлах $x = x_0$, x_1, ... Пусть *шаг* — разность между соседними значениями аргумента — постоянный и равен h. Запишем выражения для производной y'_1 в узле $x = x_1$, который слева отмечен крестиком. Используемые при этом узлы (*шаблон*) отмечены кружочками. В зависимости от способа вычисления конечных разностей получаем разные формулы для вычисления производной в одной и той же точке:

$$\bigcirc \otimes \qquad \Delta y_1 = y_1 - y_0, \quad \Delta x = h, \quad y'_1 \approx \frac{y_1 - y_0}{h} \tag{3.3}$$

с помощью *левых разностей*;

$$\otimes \bigcirc \qquad \Delta y_1 = y_2 - y_1, \quad \Delta x = h, \quad y'_1 \approx \frac{y_2 - y_1}{h} \tag{3.4}$$

с помощью *правых разностей*;

$$\bigcirc \times \bigcirc \qquad \Delta y_1 = y_2 - y_0, \quad \Delta x = 2h, \quad y_1' \approx \frac{y_2 - y_0}{2h} \tag{3.5}$$

с помощью *центральных разностей*.

Можно найти также выражения для старших производных. Например,

$$\bigcirc \otimes \bigcirc \qquad y_1'' = (y_1')' \approx \frac{y_2' - y_1'}{h} \approx \frac{(y_2 - y_1)/h - (y_1 - y_0)/h}{h} = \frac{y_2 - 2y_1 + y_0}{h^2}. \tag{3.6}$$

Таким образом, используя формулу (3.2), можно найти приближенные значения производных любого порядка. Однако при этом остается открытым вопрос о точности полученных значений. Кроме того, как будет показано ниже, для хорошей аппроксимации производной нужно использовать значения функции во многих узлах, а в формуле (3.2) это не предусмотрено.

2. Погрешность численного дифференцирования. Аппроксимируем функцию $f(x)$ некоторой функцией $\varphi(x)$, т. е. представим ее в виде

$$f(x) = \varphi(x) + R(x). \tag{3.7}$$

В качестве аппроксимирующей функции $\varphi(x)$ можно принять частичную сумму ряда или интерполяционную функцию. Тогда *погрешность аппроксимации* $R(x)$ определяется остаточным членом ряда или интерполяционной формулы.

Аппроксимирующая функция $\varphi(x)$ может быть использована также для приближенного вычисления производной функции $f(x)$. Дифференцируя равенство (3.7) необходимое число раз, можно найти значения производных $f'(x)$, $f''(x)$, ... :

$$f'(x) = \varphi'(x) + R'(x), \quad f''(x) = \varphi''(x) + R''(x), \quad \ldots$$

В качестве приближенного значения производной порядка k функции $f(x)$ можно принять значение соответствующей производной функции $\varphi(x)$, т. е. $f^{(k)}(x) \approx \varphi^{(k)}(x)$. Величина

$$R^{(k)}(x) = f^{(k)}(x) - \varphi^{(k)}(x),$$

характеризующая отклонение приближенного значения производной от ее истинного значения, называется *погрешностью аппроксимации* производной.

При численном дифференцировании функции, заданной в виде таблицы с шагом h, эта погрешность зависит от h, и ее записывают в виде $R^{(k)} = O(h^r)$ (O большое от h^r) [1]. Показатель степени r называется

[1] Напомним, это означает, что $|R^{(k)}| < Ch^r$, где $C > 0$ и не зависит от h.

порядком погрешности аппроксимации производной (или порядком точности данной аппроксимации). При этом предполагается, что значение шага по модулю меньше единицы.

Оценку погрешности легко проиллюстрировать с помощью ряда Тейлора

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)}{2!}\,\Delta x^2 + \frac{f'''(x)}{3!}\,\Delta x^3 + \dots$$

Пусть функция $f(x)$ задана в виде таблицы $f(x_i) = y_i$, $(i = 0, 1, \dots, n)$. Запишем ряд Тейлора при $x = x_1$, $\Delta x = -h$ с точностью до членов порядка h^2:

$$y_0 = y_1 - y_1' h + O(h^2).$$

Отсюда найдем значение производной в точке $x = x_1$:

$$y_1' = \frac{y_1 - y_0}{h} + O(h).$$

Это выражение совпадает с формулой (3.3), которая, как видно, является аппроксимацией первого порядка ($r = 1$). Аналогично, записывая ряд Тейлора при $\Delta x = h$, можно получить аппроксимацию (3.4). Она также имеет первый порядок.

Используем теперь ряд Тейлора для оценки погрешностей аппроксимаций (3.5) и (3.6). Полагая $\Delta x = h$ и $\Delta x = -h$ соответственно, получаем

$$\begin{aligned} y_2 &= y_1 + y_1' h + \frac{y_1''}{2!}\,h^2 + \frac{y_1'''}{3!}\,h^3 + O(h^4), \\ y_0 &= y_1 - y_1' h + \frac{y_1''}{2!}\,h^2 - \frac{y_1'''}{3!}\,h^3 + O(h^4). \end{aligned} \tag{3.8}$$

Вычитая эти равенства одно из другого, после очевидных преобразований получаем

$$y_1' = \frac{y_2 - y_0}{2h} + O(h^2).$$

Это аппроксимация производной (3.5) с помощью центральных разностей. Она имеет второй порядок.

Складывая равенства (3.8), находим оценку погрешности аппроксимации производной второго порядка вида (3.6):

$$y_1'' = \frac{y_2 - 2y_1 + y_0}{h^2} + O(h^2).$$

Таким образом, эта аппроксимация имеет второй порядок. Аналогично можно получить аппроксимации производных более высоких порядков и оценку их погрешностей.

Мы рассмотрели лишь один из источников погрешности численного дифференцирования — погрешность аппроксимации (ее также называют *погрешностью усечения*). Она определяется величиной остаточного члена.

Анализ остаточного члена нетривиален, некоторые сведения по этому вопросу приведены в п. 3. Отметим, лишь, что погрешность аппроксимации при уменьшении шага h, как правило, уменьшается.

Погрешности, возникающие при численном дифференцировании, определяются также неточными значениями функции y_i в узлах и погрешностями округлений при проведении расчетов на компьютере. Обусловленные этими причинами погрешности, в отличие от погрешности аппроксимации, возрастают с уменьшением шага h. Действительно, если при вычислении значений функции $y = f(x)$ абсолютная погрешность составляет d, то при вычислении дробей в (3.3) и (3.4) она составит $2d/h$. Поэтому суммарная погрешность численного дифференцирования может убывать при уменьшении шага лишь до некоторого предельного значения, после чего дальнейшее уменьшение шага не повысит точности результатов.

Потеря точности аппроксимации производных может быть предотвращена за счет *регуляризации* процедуры численного дифференцирования. Простейшим способом регуляризации является такой выбор шага h, при котором справедливо неравенство $|f(x+h) - f(x)| > \varepsilon$, где $\varepsilon > 0$ — некоторое малое число. При вычислении производной это исключает вычитание очень близких по величине чисел, которое обычно приводит к увеличению погрешности. Это тем более опасно при последующем делении приращения функции на малое число h. Другой способ регуляризации заключается в оценке суммарной погрешности численного дифференцирования и выборе такого шага h, который минимизировал бы эту суммарную погрешность. Возможен и еще один подход — сглаживание табличных значений функции подбором некоторой гладкой аппроксимирующей функции, например, многочлена.

3. Использование интерполяционных формул. Предположим, что функция $f(x)$, заданная в виде таблицы с постоянным шагом $h = x_i - x_{i-1}$ $(i = 1, 2, \ldots, n)$, может быть аппроксимирована интерполяционным многочленом Ньютона (2.39):

$$y \approx N(x_0 + th) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\,\Delta^2 y_0 + \ldots$$
$$\ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\,\Delta^n y_0, \quad t = \frac{x - x_0}{h}.$$

Дифференцируя этот многочлен по переменной x с учетом правила дифференцирования сложной функции:

$$\frac{dN}{dx} = \frac{dN}{dt}\frac{dt}{dx} = \frac{1}{h}\frac{dN}{dt},$$

можно получить формулы для вычисления производных любого порядка:

$$y' \approx \frac{1}{h}\left(\Delta y_0 + \frac{2t-1}{2!}\,\Delta^2 y_0 + \frac{3t^2-6t+2}{3!}\,\Delta^3 y_0 + \right.$$
$$+ \frac{4t^3-18t^2+22t-6}{4!}\,\Delta^4 y_0 +$$
$$\left. + \frac{5t^4-40t^3+105t^2-100t+24}{5!}\,\Delta^5 y_0 + \dots\right),$$

$$y'' \approx \frac{1}{h^2}\left(\Delta^2 y_0 + \frac{6t-6}{3!}\,\Delta^3 y_0 + \frac{12t^2-36t+22}{4!}\,\Delta^4 y_0 + \right.$$
$$\left. + \frac{20t^3-120t^2+210t-100}{5!}\,\Delta^5 y_0 + \dots\right),$$

. . . . . .

Число слагаемых в этих формулах зависит от количества узлов, используемых для вычисления производных. Как и при построении многочлена Ньютона, добавление к шаблону нового узла означает добавление к сумме одного слагаемого.

Пример. Вычислить в точке $x = 0.1$ первую и вторую производные функции, заданной табл. 3.1 .

Таблица 3.1

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
0	1.2833	0.5274	0.0325	0.0047	0.0002	0.0000
0.1	1.8107	0.5599	0.0372	0.0049	0.0002	
0.2	2.3606	0.5971	0.0421	0.0051		
0.3	2.9577	0.6392	0.0472			
0.4	3.5969	0.6864				
0.5	4.2833					

Здесь $h = 0.1$, $t = (0.1 - 0)/0.1 = 1$. Заполняя табл. 3.1 аналогично табл. 2.1 и используя полученные выше формулы, находим

$$y' \approx 10\left(0.5274 + \frac{2\cdot 1 - 1}{2}\cdot 0.0325 + \frac{3\cdot 1 - 6\cdot 1 + 2}{6}\cdot 0.0047 + \right.$$
$$\left. + \frac{4\cdot 1 - 18\cdot 1 + 22\cdot 1 - 6}{24}\cdot 0.0002\right) = 5.436,$$

$$y'' \approx 100\left(0.0325 + \frac{6\cdot 1 - 6}{6}\cdot 0.0047 + \frac{12 - 36 + 22}{24}\cdot 0.0002\right) = 3.25.$$

Интерполяционные многочлены Ньютона (а также Стирлинга и Бесселя) дают выражения для производных через разности $\Delta^k y$ ($k = 1, 2, \dots$). Однако на практике часто выгоднее выражать значения производных не через разности, а непосредственно через значения функции в узлах. Для получения таких формул удобно воспользоваться формулой Лагранжа с равномерным расположением узлов ($x_i - x_{i-1} = h = \text{const}$, $i = 1, 2, \dots, n$).

Запишем интерполяционный многочлен Лагранжа $L(x)$ и его остаточный член $R_L(x)$ (см. (2.34), (2.41)) для случая трех узлов интерполяции ($n = 2$) и найдем их производные:

$$
\begin{aligned}
&L(x) = \\
&= \frac{1}{2h^2}\big[(x-x_1)(x-x_2)y_0 - 2(x-x_0)(x-x_2)y_1 + (x-x_0)(x-x_1)y_2\big], \\
&R_L(x) = \frac{y'''_*}{3!}(x-x_0)(x-x_1)(x-x_2), \\
&L'(x) = \frac{1}{2h^2}\big[(2x-x_1-x_2)y_0 - 2(2x-x_0-x_2)y_1 + (2x-x_0-x_1)y_2\big], \\
&R'_L(x) = \frac{y'''_*}{3!}\big[(x-x_1)(x-x_2) + (x-x_0)(x-x_2) + (x-x_0)(x-x_1)\big].
\end{aligned}
$$

Здесь y'''_* — значение производной третьего порядка в некоторой внутренней точке $x_* \in [x_0, x_n]$.

Запишем выражение для производной y'_0 при $x = x_0$:

$$
\begin{aligned}
y'_0 &= L'(x_0) + R'_L(x_0) = \\
&= \frac{1}{2h^2}\big[(2x_0-x_1-x_2)y_0 - 2(2x_0-x_0-x_2)y_1 + (2x_0-x_0-x_1)y_2\big] + \\
&+ \frac{y'''_*}{3!}\big[(x_0-x_1)(x_0-x_2) + (x_0-x_0)(x_0-x_2) + (x_0-x_0)(x_0-x_1)\big] = \\
&= \frac{1}{2h}(-3y_0 + 4y_1 - y_2) + \frac{h^2}{3}y'''_*.
\end{aligned}
$$

Аналогичные соотношения можно получить и для значений y'_1 и y'_2 при $x = x_1, x_2$:

$$
y'_1 = \frac{1}{2h}(y_2 - y_0) - \frac{h^2}{6}y'''_*, \qquad y'_2 = \frac{1}{2h}(y_0 - 4y_1 + 3y_2) + \frac{h^2}{3}y'''_*.
$$

Для каждой из этих формул значения y'''_*, вообще говоря, различны.

Записывая интерполяционный многочлен Лагранжа и его остаточный член для случая четырех узлов ($n = 3$), получаем следующие аппроксимации производных:

$$y'_0 = \frac{1}{6h}(-11y_0 + 18y_1 - 9y_2 + 2y_3) - \frac{h^3}{4}\,y^{\mathrm{IV}}_*,$$
$$y'_1 = \frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3) + \frac{h^3}{12}\,y^{\mathrm{IV}}_*,$$
$$y'_2 = \frac{1}{6h}(y_0 - 6y_1 + 3y_2 + 2y_3) - \frac{h^3}{12}\,y^{\mathrm{IV}}_*,$$
$$y'_3 = \frac{1}{6h}(-2y_0 + 9y_1 - 18y_2 + 11y_3) + \frac{h^3}{4}\,y^{\mathrm{IV}}_*,$$

В случае пяти узлов ($n = 4$) получим

$$y'_0 = \frac{1}{12h}(-25y_0 + 48y_1 - 36y_2 + 16y_3 - 3y_4) + \frac{h^4}{5}\,y^{\mathrm{V}}_*,$$
$$y'_1 = \frac{1}{12h}(-3y_0 - 10y_1 + 18y_2 - 6y_3 + y_4) - \frac{h^4}{20}\,y^{\mathrm{V}}_*,$$
$$y'_2 = \frac{1}{12h}(y_0 - 8y_1 + 8y_3 - y_4) + \frac{h^4}{30}\,y^{\mathrm{V}}_*,$$
$$y'_3 = \frac{1}{12h}(-y_0 + 6y_1 - 18y_2 + 10y_3 + 3y_4) + \frac{h^4}{20}\,y^{\mathrm{V}}_*,$$
$$y'_4 = \frac{1}{12h}(3y_0 - 16y_1 + 36y_2 - 48y_3 + 25y_4) + \frac{h^4}{5}\,y^{\mathrm{V}}_*.$$

Таким образом, используя значения функции в $n+1$ узле, получаем аппроксимацию производных n-го порядка точности. Эти формулы можно использовать не только для узлов $x = x_0, x_1, \ldots$, но и для любых узлов $x = x_i$, $x_{i+1}, \ldots$, соответствующим образом изменяя значения индексов.

Обратим внимание на то, что при четных n наиболее простые выражения и наименьшие коэффициенты в остаточных членах получаются для производных в средних (центральных) узлах (y'_1 при $n = 2$, y'_2 при $n = 4$ и т. д.). Выпишем аппроксимации производных для узла с произвольным номером i, считая его центральным:

$$y'_i = \frac{1}{2h}(y_{i+1} - y_{i-1}) - \frac{h^2}{6}\,y'''_*, \qquad n = 2, \tag{3.9}$$
$$y'_i = \frac{1}{12h}(y_{i-2} - 8y_{i-1} + 8y_{i+1} - y_{i+2}) + \frac{h^4}{30}\,y^{\mathrm{V}}_*, \qquad n = 4.$$

Они называются *аппроксимациями производных с помощью центральных разностей* и широко используются на практике. Аппроксимация (3.9) — это не что иное, как уже встречавшаяся нам аппроксимация (3.5).

С помощью интерполяционных многочленов Лагранжа можно получить аппроксимации для старших производных. Приведем аппроксимации для вторых производных.

В случае трех узлов интерполяции ($n = 2$) имеем

$$y_0'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2) + O(h),$$
$$y_1'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2) + O(h^2),$$
$$y_2'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2) + O(h).$$

В случае четырех узлов ($n = 3$) имеем

$$y_0'' = \frac{1}{h^2}(2y_0 - 5y_1 + 4y_2 - y_3) + O(h^2),$$
$$y_1'' = \frac{1}{h^2}(y_0 - 2y_1 + y_2) + O(h^2),$$
$$y_2'' = \frac{1}{h^2}(y_1 - 2y_2 + y_3) + O(h^2),$$
$$y_3'' = \frac{1}{h^2}(-y_0 + 4y_1 - 5y_2 + 2y_3) + O(h^2).$$

В случае пяти узлов ($n = 4$) имеем

$$y_0'' = \frac{1}{12h^2}(35y_0 - 104y_1 + 114y_2 - 56y_3 + 11y_4) + O(h^3),$$
$$y_1'' = \frac{1}{12h^2}(11y_0 - 20y_1 + 6y_2 + 4y_3 - y_4) + O(h^3),$$
$$y_2'' = \frac{1}{12h^2}(-y_0 + 16y_1 - 30y_2 + 16y_3 - y_4) + O(h^4),$$
$$y_3'' = \frac{1}{12h^2}(-y_0 + 4y_1 + 6y_2 - 20y_3 + 11y_4) + O(h^3),$$
$$y_4'' = \frac{1}{12h^2}(11y_0 - 56y_1 + 114y_2 - 104y_3 + 35y_4) + O(h^3).$$

Аппроксимации вторых производных с помощью центральных разностей при четных n также наиболее выгодны.

4. Метод неопределенных коэффициентов. Аналогичные формулы можно получить и для случая произвольного расположения узлов. Использование многочлена Лагранжа в этом случае приводит к вычислению громоздких выражений, поэтому удобнее применять *метод неопределенных коэффициентов*. Он заключается в следующем. Искомое выражение для производной k-го порядка в некоторой точке $x = x_i$ представляется в виде линейной комбинации заданных значений функции в узлах $x_0, x_1, \ldots, x_n$:

$$y_i^{(k)} \approx c_0 y_0 + c_1 y_1 + \ldots + c_n y_n. \tag{3.10}$$

Предполагается, что это соотношение выполняется точно, если функция y является многочленом степени не выше n, т. е. может быть представлена

в виде

$$y = b_0 + b_1(x - x_0) + \ldots + b_n(x - x_0)^n.$$

Отсюда следует, что соотношение (3.10), в частности, должно выполняться точно для многочленов $y = 1$, $y = x - x_0$, $\ldots$, $y = (x - x_0)^n$. Подставляя последовательно эти выражения в (3.10) и требуя выполнения точного равенства, получаем систему $n + 1$ линейных алгебраических уравнений для определения неизвестных коэффициентов $c_0, c_1, \ldots, c_n$.

П р и м е р. Найти выражение для производной y_1' в случае четырех равноотстоящих узлов ($n = 3$).

Приближение (3.10) запишется в виде

$$y_1' \approx c_0 y_0 + c_1 y_1 + c_2 y_2 + c_3 y_3. \tag{3.11}$$

Используем следующие многочлены:

$$y = 1, \quad y = x - x_0, \quad y = (x - x_0)^2, \quad y = (x - x_0)^3. \tag{3.12}$$

Вычислим их производные:

$$y' = 0, \quad y' = 1, \quad y' = 2(x - x_0), \quad y' = 3(x - x_0)^2. \tag{3.13}$$

Подставляем последовательно соотношения (3.12) и (3.13) соответственно в правую и левую части (3.11) при $x = x_1$, требуя выполнения точного равенства:

$$\begin{aligned}
0 &= c_0 \cdot 1 + c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 1, \\
1 &= c_0(x_0 - x_0) + c_1(x_1 - x_0) + c_2(x_2 - x_0) + c_3(x_3 - x_0), \\
2(x_1 - x_0) &= c_0(x_0 - x_0)^2 + c_1(x_1 - x_0)^2 + c_2(x_2 - x_0)^2 + c_3(x_3 - x_0)^2, \\
3(x_1 - x_0)^2 &= c_0(x_0 - x_0)^3 + c_1(x_1 - x_0)^3 + c_2(x_2 - x_0)^3 + c_3(x_3 - x_0)^3.
\end{aligned}$$

Получаем окончательно систему уравнений в виде

$$\begin{aligned}
c_0 + c_1 + c_2 + c_3 &= 0, \\
hc_1 + 2hc_2 + 3hc_3 &= 1, \\
hc_1 + 4hc_2 + 9hc_3 &= 2, \\
hc_1 + 8hc_2 + 27hc_3 &= 3.
\end{aligned}$$

Решая эту систему, получаем

$$c_0 = -\frac{1}{3h}, \quad c_1 = -\frac{1}{2h}, \quad c_2 = \frac{1}{h}, \quad c_3 = -\frac{1}{6h}.$$

Подставляя эти значения в (3.11), находим выражение для производной:

$$y_1' \approx \frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3).$$

5. Улучшение аппроксимации. Как видно из конечно-разностных соотношений для аппроксимаций производных (см. п. 3), порядок их точности возрастает с увеличением числа узлов, используемых при аппроксимации. Однако при большом числе узлов эти соотношения становятся весьма громоздкими, что приводит к существенному возрастанию объема вычислений. Усложняется также оценка точности получаемых результатов. Вместе с тем существует простой и эффективный способ уточнения решения при фиксированном числе узлов, используемых в аппроксимирующих конечно-разностных соотношениях. Это *метод Рунге — Ромберга*. Изложим вкратце его сущность.

Пусть $F(x)$ — производная, которая подлежит аппроксимации; $f(x,h)$ — конечно-разностная аппроксимация этой производной на равномерной сетке с шагом h; R — погрешность (остаточный член) аппроксимации, главный член которой можно записать в виде $h^p\varphi(x)$, т. е.

$$R = h^p\varphi(x) + O(h^{p+1}).$$

Тогда выражение для аппроксимации производной в общем случае можно представить в виде

$$F(x) = f(x,h) + h^p\varphi(x) + O(h^{p+1}). \tag{3.14}$$

Запишем это соотношение в той же точке x при другом шаге $h_1 = kh$. Получим

$$F(x) = f(x,kh) + (kh)^p\varphi(x) + O((kh)^{p+1}). \tag{3.15}$$

Приравнивая правые части равенств (3.14) и (3.15), находим выражение для главного члена погрешности аппроксимации производной:

$$h^p\varphi(x) = \frac{f(x,h) - f(x,kh)}{k^p - 1} + O(h^{p+1}).$$

Подставляя найденное выражение в равенство (3.14), получаем *формулу Рунге*

$$F(x) = f(x,h) + \frac{f(x,h) - f(x,kh)}{k^p - 1} + O(h^{p+1}). \tag{3.16}$$

Эта формула позволяет по результатам двух расчетов значений производной $f(x,h)$ и $f(x,kh)$ (с шагами h и kh) с порядком точности p найти ее уточненное значение с порядком точности $p+1$.

Пример. Вычислить производную функции $y = x^3$ в точке $x = 1$.

Очевидно, что $y' = 3x^2$; поэтому $y'(1) = 3$. Найдем теперь эту производную численно. Составим таблицу значений функции:

x	0.8	0.9	1.0
y	0.512	0.729	1.0

Воспользуемся аппроксимацией производной с помощью левых разностей, имеющей первый порядок ($p = 1$). Примем шаг равным 0.1 и 0.2, т. е. $k = 2$. Получим

$$f(x,h) = y'(1,0.1) = \frac{f(1) - f(0.9)}{0.1} = \frac{1 - 0.729}{0.1} = 2.71,$$
$$f(x,kh) = y'(1,0.2) = \frac{f(1) - f(0.8)}{0.2} = \frac{1 - 0.512}{0.2} = 2.44.$$

По формуле Рунге найдем уточненное значение производной:

$$F(x) = y'(1) \approx 2.71 + \frac{2.71 - 2.44}{2^1 - 1} = 2.98.$$

Таким образом, формула Рунге дает более точное значение производной. В общем случае порядок точности аппроксимации увеличивается на единицу.

Мы рассмотрели уточнение решения, полученного при двух значениях шага. Предположим теперь, что расчеты могут быть проведены с шагами $h_1, h_2, \ldots, h_q$. Тогда можно получить уточненное решение для производной $F(x)$ по *формуле Ромберга*, которая имеет вид

$$F(x) = \frac{\begin{vmatrix} f(x,h_1) & h_1^p & h_1^{p+1} & \ldots & h_1^{p+q-2} \\ f(x,h_2) & h_2^p & h_2^{p+1} & \ldots & h_2^{p+q-2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ f(x,h_q) & h_q^p & h_q^{p+1} & \ldots & h_q^{p+q-2} \end{vmatrix}}{\begin{vmatrix} 1 & h_1^p & h_1^{p+1} & \ldots & h_1^{p+q-2} \\ 1 & h_2^p & h_2^{p+1} & \ldots & h_2^{p+q-2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & h_q^p & h_q^{p+1} & \ldots & h_q^{p+q-2} \end{vmatrix}} + O(h^{p+q-1}). \qquad (3.17)$$

Таким образом, порядок точности возрастает на $q - 1$. Заметим, что для успешного применения уточнения исходная функция должна иметь непрерывные производные достаточно высокого порядка.

6. Частные производные. Рассмотрим функцию двух переменных $u = f(x,y)$, заданную в табличном виде: $u_{ij} = f(x_i, y_j)$, где $x_i = x_0 + ih_1$ $(i = 0,1,\ldots,I)$, $y_j = y_0 + jh_2$ $(j = 0,1,\ldots,J)$. В табл. 3.2 представлена часть данных, которые нам в дальнейшем понадобятся.

Используя понятие частной производной, можем приближенно записать для малых значений шагов h_1 и h_2

$$\frac{\partial u}{\partial x} \approx \frac{f(x+h_1,y) - f(x,y)}{h_1}, \qquad \frac{\partial u}{\partial y} \approx \frac{f(x,y+h_2) - f(x,y)}{h_2}.$$

Воспользовавшись введенными выше обозначениями, получим следующие приближенные выражения (аппроксимации) для частных производных

Таблица 3.2

y \ x	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
y_{j-2}	$u_{i-2,\,j-2}$	$u_{i-1,\,j-2}$	$u_{i,\,j-2}$	$u_{i+1,\,j-2}$	$u_{i+2,\,j-2}$
y_{j-1}	$u_{i-2,\,j-1}$	$u_{i-1,\,j-1}$	$u_{i,\,j-1}$	$u_{i+1,\,j-1}$	$u_{i+2,\,j-1}$
y_j	$u_{i-2,\,j}$	$u_{i-1,\,j}$	u_{ij}	$u_{i+1,\,j}$	$u_{i+2,\,j}$
y_{j+1}	$u_{i-2,\,j+1}$	$u_{i-1,\,j+1}$	$u_{i,\,j+1}$	$u_{i+1,\,j+1}$	$u_{i+2,\,j+1}$
y_{j+2}	$u_{i-2,\,j+2}$	$u_{i-1,\,j+2}$	$u_{i,\,j+2}$	$u_{i+1,\,j+2}$	$u_{i+2,\,j+2}$

в узле (x_i, y_j) с помощью отношений конечных разностей:

$$\left(\frac{\partial u}{\partial x}\right)_{ij} \approx \frac{u_{i+1,j} - u_{ij}}{h_1}\,, \qquad \left(\frac{\partial u}{\partial y}\right)_{ij} \approx \frac{u_{i,j+1} - u_{ij}}{h_2}\,.$$

Для численного дифференцирования функций многих переменных можно, как и ранее, использовать интерполяционные многочлены. Однако рассмотрим здесь другой способ — разложение в ряд Тейлора функции двух переменных:

$$f(x+\Delta x,\, y+\Delta y) = f(x,y) + \frac{\partial f}{\partial x}\,\Delta x + \frac{\partial f}{\partial y}\,\Delta y + \\ + \frac{1}{2!}\left(\frac{\partial^2 f}{\partial x^2}\,\Delta x^2 + 2\frac{\partial^2 f}{\partial x\,\partial y}\,\Delta x\Delta y + \frac{\partial^2 f}{\partial y^2}\,\Delta y^2\right) + \\ + \frac{1}{3!}\left(\frac{\partial^3 f}{\partial x^3}\,\Delta x^3 + 3\frac{\partial^3 f}{\partial x^2\,\partial y}\,\Delta x^2\Delta y + 3\frac{\partial^3 f}{\partial x\,\partial y^2}\,\Delta x\Delta y^2 + \frac{\partial^3 f}{\partial y^3}\,\Delta y^3\right) + \ldots \tag{3.18}$$

Используем эту формулу дважды:
1) найдем $u_{i+1,j} = f(x_i + h_1, y_j)$ при $\Delta x = h_1,\ \Delta y = 0$;
2) найдем $u_{i-1,j} = f(x_i - h_1, y_j)$ при $\Delta x = -h_1,\ \Delta y = 0$.
Получим

$$u_{i+1,j} = u_{ij} + \left(\frac{\partial u}{\partial x}\right)_{ij} h_1 + \frac{1}{2!}\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} h_1^2 + \frac{1}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right)_{ij} h_1^3 + \ldots,$$

$$u_{i-1,j} = u_{ij} - \left(\frac{\partial u}{\partial x}\right)_{ij} h_1 + \frac{1}{2!}\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} h_1^2 - \frac{1}{3!}\left(\frac{\partial^3 u}{\partial x^3}\right)_{ij} h_1^3 + \ldots$$

Вычитая почленно из первого равенства второе, получаем

$$u_{i+1,j} - u_{i-1,j} = 2h_1\left(\frac{\partial u}{\partial x}\right)_{ij} + O(h_1^3).$$

Отсюда найдем аппроксимацию производной с помощью центральных разностей:

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h_1} + O(h_1^2).$$

Она имеет второй порядок.

Аналогично могут быть получены аппроксимации производной $\partial u/\partial y$, а также старших производных. В частности, для второй производной можно получить

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_1^2} + O(h_1^2).$$

Записывая разложения в ряд (3.18) при разных значениях Δx и Δy, можно вывести формулы численного дифференцирования с необходимым порядком аппроксимации.

Приведем окончательные формулы для некоторых аппроксимаций частных производных. Слева указывается используемый шаблон. Значения производных вычисляются в узле (x_i, y_j), отмеченном крестиком (напомним, что на шаблонах и в табл. 3.2 по горизонтали изменяются переменная x и индекс i, по вертикали — переменная y и индекс j):

○ × ○ $\left(\dfrac{\partial u}{\partial x}\right)_{ij} \approx \dfrac{u_{i+1,j} - u_{i-1,j}}{2h_1}$,

○ / × / ○ $\left(\dfrac{\partial u}{\partial y}\right)_{ij} \approx \dfrac{u_{i,j+1} - u_{i,j-1}}{2h_2}$,

○ ⊗ ○ $\left(\dfrac{\partial^2 u}{\partial x^2}\right)_{ij} \approx \dfrac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_1^2}$,

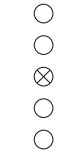
○ / ⊗ / ○ $\left(\dfrac{\partial^2 u}{\partial y^2}\right)_{ij} \approx \dfrac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_2^2}$,

○ ○ / × / ○ ○ $\left(\dfrac{\partial^2 u}{\partial x \partial y}\right)_{ij} \approx \dfrac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{4h_1 h_2}$,

○ ○ / × / ○ ○ $\left(\dfrac{\partial u}{\partial x}\right)_{ij} \approx \dfrac{u_{i+1,j+1} - u_{i-1,j+1} + u_{i+1,j-1} - u_{i-1,j-1}}{4h_1}$,

○ ○ / × / ○ ○ $\left(\dfrac{\partial u}{\partial y}\right)_{ij} \approx \dfrac{u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j+1} - u_{i-1,j-1}}{4h_2}$,

○ ○ ⊗ ○ ○ $\left(\dfrac{\partial^2 u}{\partial x^2}\right)_{ij} \approx \dfrac{-u_{i+2,j} + 16u_{i+1,j} - 30u_{ij} + 16u_{i-1,j} - u_{i-2,j}}{12h_1^2}$,



$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{ij} \approx \frac{-u_{i,j+2} + 16u_{i,j+1} - 30u_{ij} + 16u_{i,j-1} - u_{i,j-2}}{12h_2^2},$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} \approx \frac{1}{3h_1^2}(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} + u_{i+1,j} - 2u_{ij} + u_{i-1,j} + u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1}),$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right)_{ij} \approx \frac{1}{3h_2^2}(u_{i+1,j+1} - 2u_{i+1,j} + u_{i+1,j-1} + u_{i,j+1} - 2u_{ij} + u_{i,j-1} + u_{i-1,j+1} - 2u_{i-1,j} + u_{i-1,j-1}).$$

Приведенные аппроксимации производных могут быть использованы при построении разностных схем для решения уравнений с частными производными (см. гл. 8).

§ 2. Численное интегрирование

1. Вводные замечания. Напомним некоторые понятия, необходимые для дальнейшего изложения.

Пусть на отрезке $[a, b]$ задана функция $y = f(x)$. С помощью точек $x_0, x_1, \ldots, x_n$ разобьем отрезок $[a, b]$ на n элементарных отрезков $[x_{i-1}, x_i]$ $(i = 1, 2, \ldots, n)$, причем $x_0 = a$, $x_n = b$. На каждом из этих отрезков выберем произвольную точку ξ_i $(x_{i-1} \leqslant \xi_i \leqslant x_i)$ и найдем произведение s_i значения функции в этой точке $f(\xi_i)$ на длину элементарного отрезка $\Delta x_i = x_i - x_{i-1}$:

$$s_i = f(\xi_i)\Delta x_i. \tag{3.19}$$

Составим сумму всех таких произведений:

$$S_n = s_1 + s_2 + \ldots + s_n = \sum_{i=1}^{n} f(\xi_i)\Delta x_i. \tag{3.20}$$

Сумма S_n называется *интегральной суммой*. *Определенным интегралом* от функции $f(x)$ на отрезке $[a, b]$ называется предел интегральной суммы при таком неограниченном увеличении числа точек разбиения, при котором длина наибольшего из элементарных отрезков стремится к нулю:

$$\int_a^b f(x)\,dx = \lim_{\max \Delta x_i \to 0} \sum_{i=1}^{n} f(\xi_i)\Delta x_i. \tag{3.21}$$

Т е о р е м а (существования определенного интеграла). *Если функция $f(x)$ непрерывна на $[a,b]$, то предел интегральной суммы существует и не зависит ни от способа разбиения отрезка $[a,b]$ на элементарные отрезки, ни от выбора точек ξ_i.*

Геометрический смысл введенных понятий для случая $f(x)>0$ проиллюстрирован на рис. 3.1. Абсциссами точек M_i являются значения ξ_i, ординатами — значения $f(\xi_i)$. Выражения (3.19) при $i=1,2,\ldots,n$ описывают площади элементарных прямоугольников (штриховые линии), интегральная сумма (3.20) — площадь ступенчатой фигуры, образуемой этими прямоугольниками. При неограниченном увеличении числа точек деления и стремлении к нулю всех элементов Δx_i верхняя граница фигуры (ломаная) переходит в линию $y=f(x)$. Площадь полученной фигуры, которую называют *криволинейной трапецией*, равна определенному интегралу (3.21).

Во многих случаях, когда подынтегральная функция задана в аналитическом виде, определенный интеграл удается вычислить непосредственно с помощью неопределенного интеграла (вернее, первообразной) по *формуле Ньютона—Лейбница*. Она состоит в том, что определенный интеграл равен приращению первообразной $F(x)$ на отрезке интегрирования:

$$\int_a^b f(x)\,dx = F(x)\Big|_a^b = F(b)-F(a). \tag{3.22}$$

Однако на практике этой формулой часто нельзя воспользоваться по двум основным причинам:

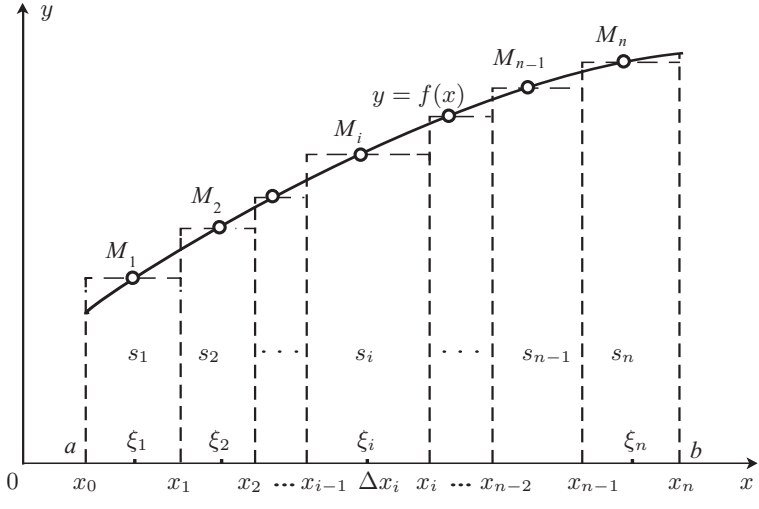


Рис. 3.1

1) вид функции $f(x)$ не допускает непосредственного интегрирования, т. е. первообразную нельзя выразить в элементарных функциях;

2) значения функции $f(x)$ заданы только на фиксированном конечном множестве точек x_i, т. е. функция задана в виде таблицы.

В этих случаях используются приближенные методы интегрирования. Они основаны на аппроксимации подынтегральной функции некоторыми более простыми выражениями, например многочленами.

Одним из таких способов, который может быть использован для вычисления интегралов в первом случае, является *представление подынтегральной функции в виде степенного ряда* (ряда Тейлора). Это позволяет свести вычисление интеграла от сложной функции к интегрированию многочлена, представляющего первые несколько членов ряда.

П р и м е р. Вычислить интеграл $I = \int\limits_0^1 e^{-x^2}\,dx$ с погрешностью 10^{-4}.

Воспользуемся разложением экспоненты в ряд:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

Используя последнее выражение и заменяя x на $-x^2$, записываем интеграл в виде

$$I = \int\limits_0^1 \left(1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \dots\right) dx =$$

$$= x - \frac{x^3}{3} + \frac{x^5}{5 \cdot 2!} - \frac{x^7}{7 \cdot 3!} + \dots \Big|_0^1 = 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} + \dots \approx 0.7468.$$

Более универсальными методами, которые пригодны для обоих случаев, являются *методы численного интегрирования*, основанные на аппроксимации подынтегральной функции с помощью интерполяционных многочленов. Такая аппроксимация позволяет приближенно заменить определенный интеграл конечной суммой

$$\int\limits_a^b f(x)\,dx \approx \sum_{i=0}^{n} \alpha_i y_i, \tag{3.23}$$

где y_i — значения функции в узлах интерполяции, α_i — числовые коэффициенты. Соотношение (3.23) называется *квадратурной формулой*, а его правая часть — *квадратурной суммой*. В зависимости от способа ее вычисления получаются разные методы численного интегрирования (квадратурные формулы) — методы прямоугольников, трапеций, парабол, сплайнов и др.

Квадратурную сумму можно вычислить по аналогии с интегральной суммой (3.20)

$$\sum_{i=0}^{n} \alpha_i y_i = \sum_{i=1}^{n} \sigma_i,$$

где σ_i — приближенное значение площади элементарной криволинейной трапеции, соответствующей элементарному отрезку $[x_{i-1}, x_i]$. Например, можно положить $\sigma_i = s_i$ при некотором выборе точки ξ_i в (3.19). В дальнейшем при вычислении квадратурной суммы будем аппроксимировать подынтегральную функцию, используя кусочную (локальную) интерполяцию.

Следует отметить, что к вычислению определенного интеграла сводятся многие практические задачи: вычисление площади фигур, определение работы переменной силы и т. д. Решение задач с использованием кратных интегралов также обычно может быть в конечном итоге сведено к вычислению определенных интегралов.

2. Методы прямоугольников и трапеций. Простейшим методом численного интегрирования является *метод прямоугольников*. Он непосредственно использует замену определенного интеграла интегральной суммой (3.20). В качестве точек ξ_i могут выбираться левые ($\xi_i = x_{i-1}$) или правые ($\xi_i = x_i$) границы элементарных отрезков. Обозначая $f(x_i) = y_i$, $\Delta x_i = h_i$, получаем следующие *формулы метода прямоугольников* соответственно для этих двух случаев:

$$\int_a^b f(x)\,dx \approx h_1 y_0 + h_2 y_1 + \ldots + h_n y_{n-1}, \tag{3.24}$$

$$\int_a^b f(x)\,dx \approx h_1 y_1 + h_2 y_2 + \ldots + h_n y_n. \tag{3.25}$$

Широко распространенным и более точным является вид формулы прямоугольников, использующий значения функции в средних точках элементарных отрезков (в *полуцелых узлах*):

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} h_i f(x_{i-1/2}), \tag{3.26}$$

$$x_{i-1/2} = (x_{i-1} + x_i)/2 = x_{i-1} + h_i/2, \quad i = 1, 2, \ldots, n.$$

В дальнейшем под методом прямоугольников будем понимать последний алгоритм (он еще называется *методом средних*).

В рассмотренных методах прямоугольников используется *кусочно постоянная* интерполяция: на каждом элементарном отрезке функция $f(x)$

приближается функцией, принимающей постоянные значения (константой). При этом площадь всей фигуры (криволинейной трапеции) приближенно складывается из площадей элементарных прямоугольников. На рис. 3.2 верхняя, средняя и нижняя горизонтальные штриховые линии относятся к элементарным прямоугольникам, которые соответствуют формулам (3.25), (3.26) и (3.24).

Метод трапеций использует линейную интерполяцию, т. е. график функции $y = f(x)$ представляется в виде ломаной, соединяющей точки (x_i, y_i). В этом случае площадь всей фигуры приближенно складывается из площадей элементарных прямолинейных трапеций (рис. 3.2). Площадь каждой такой трапеции равна произведению полусуммы оснований на высоту:

$$\sigma_i = \frac{y_{i-1} + y_i}{2}\, h_i, \quad i = 1, 2, \ldots, n.$$

Складывая все эти равенства, получаем формулу трапеций для численного интегрирования:

$$\int_a^b f(x)\,dx \approx \frac{1}{2}\sum_{i=1}^{n} h_i(y_{i-1} + y_i). \qquad (3.27)$$

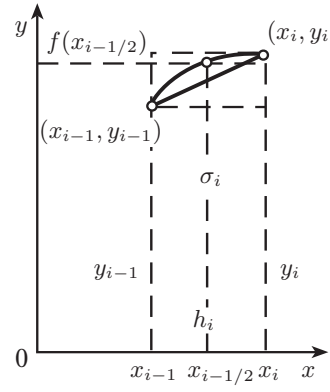


Рис. 3.2. Вычисление σ_i в методах прямоугольников и трапеций

Важным частным случаем рассмотренных формул является их применение при численном интегрировании с постоянным шагом $h_i = h = \text{const}$ $(i = 1, 2, \ldots, n)$. Формулы прямоугольников и трапеций в этом случае принимают соответственно вид

$$\int_a^b f(x)\,dx \approx h\sum_{i=1}^{n} f(x_{i-1/2}), \qquad (3.28)$$

$$\int_a^b f(x)\,dx \approx h\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right). \qquad (3.29)$$

Рассмотрим пример использования этих формул при ручном счете для простейшего интеграла, допускающего также непосредственное вычисление. Такой пример позволит сравнить результаты расчетов, полученные различными способами.

Пример. Вычислить интеграл $I = \int_0^1 \frac{dx}{1 + x^2}$.

Этот интеграл легко вычисляется по формуле (3.22):

$$I = \operatorname{arctg} x|_0^1 = \frac{\pi}{4} \approx 0.785398.$$

Используем теперь для вычисления данного интеграла формулы прямоугольников и трапеций. Разобьем отрезок интегрирования $[0, 1]$ на десять равных частей: $n = 10$, $h = 0.1$. Вычислим значения подынтегральной функции $y_i = 1/(1 + x_i^2)$ в точках разбиения $x_i = x_{i-1} + h$, а также в полуцелых точках $x_{i-1/2} = x_{i-1} + h/2$ $(i = 1, 2, \ldots, 10)$ (табл. 3.3).

Таблица 3.3

i	x_i	y_i	$x_{i-1/2}$	$y_{i-1/2}$
0	0.0	1.000000		
1	0.1	0.990099	0.05	0.997506
2	0.2	0.961538	0.15	0.977995
3	0.3	0.917431	0.25	0.941176
4	0.4	0.862069	0.35	0.890868
5	0.5	0.800000	0.45	0.831601
6	0.6	0.735294	0.55	0.767754
7	0.7	0.671141	0.65	0.702988
8	0.8	0.609756	0.75	0.640000
9	0.9	0.552486	0.85	0.580552
10	1.0	0.500000	0.95	0.525624

По формуле прямоугольников (3.28) получим

$$I_1 = h \sum_{i=1}^{10} y_{i-1/2} = 0.1 \cdot (0.997506 + \ldots + 0.525624) = 0.785606.$$

Погрешность в вычислении интеграла составляет $\Delta I = I - I_1 = -0.00021$ (около $0.027\,\%$). Используя формулу трапеций (3.29), находим

$$I_2 = 0.1 \cdot (0.750000 + 0.990099 + \ldots + 0.552486) = 0.784981.$$

Погрешность здесь равна $\Delta I_2 = 0.00042$ (около $0.054\,\%$).

Таким образом, в рассмотренном примере лучшую точность вычисления интеграла дает формула прямоугольников. Это, на первый взгляд, неожиданный результат, поскольку формула прямоугольников использует интерполяцию нулевого порядка (кусочно – постоянную), в то время как формула трапеций использует кусочно – линейную интерполяцию. Повышение точности здесь объясняется способом вычисления элементарных

площадей σ_i, использующим значения функции в центральной точке $x_{i-1/2}$ отрезка $[x_{i-1}, x_i]$. Заметим, что использование формул прямоугольников в виде (3.24) или (3.25) приведет к погрешности более 3 %.

В п. 5 показано, что погрешность численного интегрирования определяется шагом разбиения. Уменьшая этот шаг, можно добиться большей точности. Правда, увеличивать число точек не всегда возможно. Если функция задана в табличном виде, приходится, как правило, ограничиваться данным множеством точек. Повышение точности может быть в этом случае достигнуто за счет повышения степени используемых интерполяционных многочленов. Рассмотрим два таких способа численного интегрирования: использование квадратичной интерполяции (метод Симпсона) и интерполирование с помощью сплайнов.

3. Метод Симпсона. Разобьем отрезок интегрирования $[a, b]$ на четное число n равных частей с шагом h. На каждом отрезке $[x_0, x_2]$, $[x_2, x_4]$, ... , $[x_{i-1}, x_{i+1}]$, ... , $[x_{n-2}, x_n]$ подынтегральную функцию $f(x)$ заменим интерполяционным многочленом второй степени:

$$f(x) \approx \varphi_i(x) = a_i x^2 + b_i x + c_i, \qquad x_{i-1} \leqslant x \leqslant x_{i+1}.$$

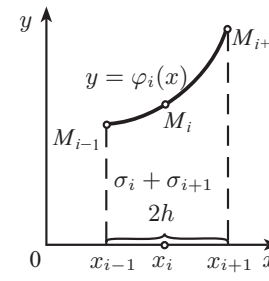


Рис. 3.3

Коэффициенты этих квадратных трехчленов могут быть найдены из условий равенства многочлена в точках x_i соответствующим табличным данным y_i. В качестве $\varphi_i(x)$ можно принять интерполяционный многочлен Лагранжа второй степени, проходящий через точки $M_{i-1}(x_{i-1}, y_{i-1})$, $M_i(x_i, y_i)$, $M_{i+1}(x_{i+1}, y_{i+1})$:

$$\varphi_i(x) = \frac{(x - x_i)(x - x_{i+1})}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})}\, y_{i-1} + \\ + \frac{(x - x_{i-1})(x - x_{i+1})}{(x_i - x_{i-1})(x_i - x_{i+1})}\, y_i + \frac{(x - x_{i-1})(x - x_i)}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)}\, y_{i+1}.$$

Сумма элементарных площадей σ_i и σ_{i+1} (рис. 3.3) может быть вычислена с помощью определенного интеграла. Учитывая равенства $x_{i+1} - x_i = x_i - x_{i-1} = h$, получаем

$$\sigma_i + \sigma_{i+1} = \int\limits_{x_{i-1}}^{x_{i+1}} \varphi_i(x)\, dx = \frac{1}{2h^2} \int\limits_{x_{i-1}}^{x_{i+1}} [(x - x_i)(x - x_{i+1})\, y_{i-1} - \\ - 2(x - x_{i-1})(x - x_{i+1})\, y_i + (x - x_{i-1})(x - x_i) y_{i+1}]\, dx = \\ = \frac{h}{3}\,(y_{i-1} + 4y_i + y_{i+1}).$$

Проведя такие вычисления для каждого элементарного отреза $[x_{i-1}, x_{i+1}]$, просуммируем полученные выражения:

$$S = \frac{h}{3}\,(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots + 2y_{n-2} + 4y_{n-1} + y_n).$$

Данное выражение для S принимается в качестве значения определенного интеграла:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\,[(y_0 + 4(y_1 + y_3 + \ldots + y_{n-1}) + \\ + 2(y_2 + y_4 + \ldots + y_{n-2}) + y_n]\,. \quad (3.30)$$

Полученное соотношение называется *формулой Симпсона* или *формулой парабол*.

Эту формулу можно получить и другими способами, например двукратным применением метода трапеций при разбиениях отрезка $[a, b]$ на части с шагами h и $2h$ или комбинированием формул прямоугольников и трапеций (см. п. 5).

Иногда формулу Симпсона записывают с применением полуцелых индексов. В этом случае число отрезков разбиения n произвольно (не обязательно четно), и формула Симпсона имеет вид

$$\int_a^b f(x)\,dx \approx \frac{h}{6}\,[(y_0 + 4(y_{1/2} + y_{3/2} + \ldots + y_{n-1/2}) + \\ + 2(y_1 + y_2 + \ldots + y_{n-1}) + y_n]\,. \quad (3.31)$$

Легко видеть, что формула (3.31) совпадет с (3.30), если формулу (3.30) применить для числа отрезков разбиения $2n$ и шага $h/2$.

П р и м е р. Вычислить по методу Симпсона интеграл $I = \int_0^1 \frac{dx}{1+x^2}$. Значения функции при $n = 10$, $h = 0.1$ приведены в табл. 3.3.

Применяя формулу (3.30), находим

$$I = \frac{0.1}{3}\,[y_0 + 4(y_1 + y_3 + y_5 + y_7 + y_9) + \\ + 2(y_2 + y_4 + y_6 + y_8) + y_{10}] = \ldots = 0.785398.$$

Результат численного интегрирования с использованием метода Симпсона оказался совпадающим с точным значением (шесть значащих цифр).

Один из возможных алгоритмов вычисления определенного интеграла по методу Симпсона представлен на рис. 3.4. В качестве исходных данных задаются границы отрезка интегрирования $[a, b]$, погрешность ε, а также формула для вычисления значений подынтегральной функции $y = f(x)$.

Первоначально отрезок $[a, b]$ разбивается на две части с шагом $h = (b-a)/2$. Вычисляется значение интеграла I_1. Потом число шагов удваивается, вычисляется значение I_2 с шагом $h/2$. Условие окончание счета принимается в виде $|I_1 - I_2| < \varepsilon$. Если это условие не выполнено, происходит новое деление шага пополам и т. д.

Отметим, что представленный на рис. 3.4 алгоритм не является оптимальным: при вычислении каждого приближения I_2 не используются значения функции $f(x)$, уже найденные на предыдущем этапе. Более экономичные алгоритмы будут рассмотрены в п. 6.

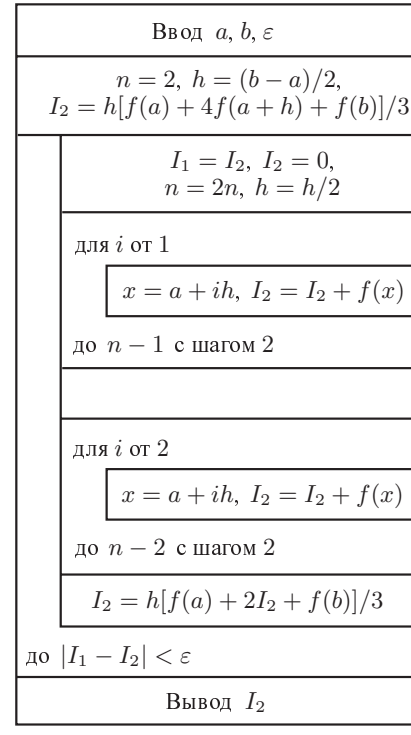


Рис. 3.4. Метод Симпсона

4. Использование сплайнов. Одним из методов численного интегрирования, особенно эффективным при строго ограниченном числе узлов, является *метод сплайнов*, использующий интерполяцию сплайнами (см. гл. 2, § 3, п. 5).

Разобьем отрезок интегрирования $[a, b]$ на n частей точками x_i. Пусть $\Delta x_i = h_i$ $(i = 1, 2, \ldots, n)$. На каждом элементарном отрезке интерполируем подынтегральную функцию $f(x)$ с помощью кубического сплайна:

$$\varphi_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \quad x_{i-1} \leqslant x \leqslant x_i, \quad i = 1, 2, \ldots, n. \tag{3.32}$$

Выражение для интеграла представим в виде

$$I = \int_a^b f(x)\,dx = \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} f(x)\,dx \approx \sum_{i=1}^{n} \int_{x_{i-1}}^{x_i} \varphi(x)\,dx. \tag{3.33}$$

Используя выражение (3.32), в результате вычисления интегралов находим

$$I \approx \sum_{i=1}^{n} \left(a_i h_i + \frac{1}{2}\, b_i h_i^2 + \frac{1}{3}\, c_i h_i^3 + \frac{1}{4}\, d_i h_i^4 \right). \tag{3.34}$$

Способ вычисления коэффициентов a_i, b_i, c_i, d_i описан в гл. 2. Здесь лишь отметим, что $a_i = y_{i-1}$.

Подставив в (3.34) выражения для a_i, b_i, d_i, эту формулу можно представить в удобном для практических расчетов виде

$$I \approx \frac{1}{2}\sum_{i=1}^{n} h_i(y_{i-1}+y_i) - \frac{1}{12}\sum_{i=1}^{n} h_i^3(c_i+c_{i+1}). \tag{3.35}$$

Отметим, что во всех предыдущих методах (см. п. 2, 3) формулы численного интегрирования можно записать в виде линейной комбинации табличных значений функции, т. е. в виде квадратурной формулы (3.23) с постоянными коэффициентами α_i. При использовании сплайнов такое представление невозможно, поскольку коэффициенты α_i зависят в этом случае от всех значений y_i.

5. Погрешность численного интегрирования. При вычислении приближенного значения интеграла по квадратурной формуле (3.23) допускается погрешность

$$R = \int_a^b f(x)\,dx - \sum_{i=0}^{n} \alpha_i y_i.$$

Она зависит от шага разбиения, и ее можно представить в виде $R = O(h^r)$. В случае переменного шага можно принять $h = \max h_i$, $h_i = \Delta x_i$. Как и в случае численного дифференцирования, показатель степени r называют *порядком точности* данной квадратурной формулы (или данного метода). Квадратурная формула должна быть составлена таким образом, чтобы для любой интегрируемой на отрезке $[a,b]$ функции $f(x)$ при $h \to 0$ ($n \to \infty$) значения интеграла, получаемые путем численного интегрирования, сходились к его точному значению. Это означает выполнение неравенства $r > 0$.

Погрешность R, допускаемую при интегрировании функции по отрезку $[a,b]$, можно представить в виде суммы погрешностей r_i, допускаемых на каждом элементарном отрезке:

$$R = \sum_{i=1}^{n} r_i, \quad r_i = \int_{x_{i-1}}^{x_i} f(x)\,dx - \sigma_i. \tag{3.36}$$

Получим выражения для погрешностей формул прямоугольников и трапеций. Запишем разложение функции $y = f(x)$ в ряд Тейлора на отрезке $[x_{i-1}, x_i]$:

$$f(x) = y_{i-1/2} + y'_{i-1/2}(x - x_{i-1/2}) + \frac{1}{2!}\,y''_{i-1/2}(x - x_{i-1/2})^2 + \\ + \frac{1}{3!}\,y'''_{i-1/2}(x - x_{i-1/2})^3 + O(h_i^4). \tag{3.37}$$

Заметим, что разложение (3.37) можно оборвать на k-м слагаемом, если в этом слагаемом производную $y^{(k)}_{i-1/2}$ заменить на $f^{(k)}(x_*)$, где x_* —

некоторая точка отрезка $[x_{i-1}, x_i]$. Например,

$$f(x) = y_{i-1/2} + y'_{i-1/2}(x - x_{i-1/2}) + \frac{1}{2!} f''(x_*)(x - x_{i-1/2})^2.$$

Проинтегрировав обе части равенства (3.37) по отрезку $[x_{i-1}, x_i]$, получим

$$\int_{x_{i-1}}^{x_i} f(x)\,dx = y_{i-1/2}\Big|_{x_{i-1}}^{x_i} + \frac{1}{2}\, y'_{i-1/2}(x - x_{i-1/2})^2\Big|_{x_{i-1}}^{x_i} +$$
$$+ \frac{1}{6}\, y''_{i-1/2}(x - x_{i-1/2})^3\Big|_{x_{i-1}}^{x_i} + \frac{1}{24}\, y'''_{i-1/2}(x - x_{i-1/2})^4\Big|_{x_{i-1}}^{x_i} + O(h_i^5) =$$
$$= y_{i-1/2}h_i + \frac{h_i^3}{24}\, y''_{i-1/2} + O(h_i^5). \quad (3.38)$$

Для метода прямоугольников $y_{i-1/2}h_i = \sigma_i$; отсюда

$$r_{\text{пр},i} = \frac{h_i^3}{24}\, y''_{i-1/2} + O(h_i^5) = \frac{h_i^3}{24}\, f''(x_*). \quad (3.39)$$

Для метода трапеций $\sigma_i = (y_{i-1} + y_i)h_i/2$. Вычислим σ_i, для чего найдем y_{i-1} и y_i из (3.37) (подставив в (3.37) $x = x_{i-1}$ и $x = x_i$ соответственно), сложим полученные выражения и домножим их сумму на $h_i/2$:

$$\sigma_i = \frac{y_{i-1} + y_i}{2}\, h_i = y_{i-1/2}h_i + \frac{h_i^3}{8}\, y''_{i-1/2} + O(h_i^5).$$

Из этого соотношения выразим $y_{i-1/2}h_i$ и подставим в (3.38):

$$\int_{x_{i-1}}^{x_i} f(x)\,dx = \frac{y_{i-1} + y_i}{2}\, h_i - \frac{h_i^3}{12}\, y''_{i-1/2} + O(h_i^5).$$

Отсюда

$$r_{\text{тр},i} = -\frac{h_i^3}{12}\, y''_{i-1/2} + O(h_i^5) = -\frac{h_i^3}{12}\, f''(x_*). \quad (3.40)$$

Выражение для погрешности метода трапеций (3.40) можно получить и другим способом, проинтегрировав по отрезку $[x_{i-1}, x_i]$ интерполяционный многочлен с учетом остаточного члена (2.41).

Используя формулы прямоугольников и трапеций, можно получить *уточненные значения интегралов*, если учесть характер погрешностей этих формул. Как следует из (3.39) и (3.40), главный член погрешности формулы трапеций вдвое больше по модулю и имеет другой знак. На основании этого можно записать уточненную формулу для вычисления определенного

интеграла с использованием значений $I_{\text{пр}}$ и $I_{\text{тр}}$, вычисленных по методам прямоугольников и трапеций:

$$I \approx (2I_{\text{пр}} + I_{\text{тр}})/3. \tag{3.41}$$

Погрешность полученной формулы составит

$$r_i = (2r_{\text{пр},i} + r_{\text{тр},i})/3 = O(h_i^5).$$

Нетрудно убедиться, что формула (3.41) совпадает с (3.31) — формулой Симпсона, записанной с помощью полуцелых индексов.

Предположим теперь шаг разбиения постоянным, $h_i = h = \text{const}$ ($i = 1, 2, \ldots, n$), и оценим погрешность метода прямоугольников на отрезке $[a, b]$ согласно (3.36) и (3.39):

$$|R_{\text{пр}}| = \Big|\sum_{i=1}^{n} r_{\text{пр},i}\Big| = \frac{h^3}{24}\Big|\sum_{i=1}^{n} f''(x_{*,i})\Big| \leqslant \frac{h^3}{24}\sum_{i=1}^{n}|f''(x_{*,i})| \leqslant$$

$$\leqslant \frac{h^3}{24}\sum_{i=1}^{n} M_2 = \frac{h^3}{24}\,nM_2 = \frac{(b-a)h^2}{24}\,M_2, \tag{3.42}$$

где

$$M_2 = \max_{a \leqslant x \leqslant b}\big|f''(x)\big|.$$

При выводе (3.42) мы воспользовались неравенством

$$|a_1 + a_2 + \ldots + a_n| \leqslant |a_1| + |a_2| + \ldots + |a_n|.$$

Аналогично (3.42), для метода трапеций имеем

$$|R_{\text{тр}}| \leqslant \frac{(b-a)h^2}{12} M_2.$$

Для метода Симпсона (3.30) можно получить следующую оценку погрешности:

$$|R_{\text{C}}| \leqslant \frac{(b-a)h^4}{180} M_4,$$

где M_4 — максимум модуля четвертой производной функции $f(x)$.

Сравнив методы прямоугольников и трапеций с методом Симпсона, отметим, что последний обладает более высоким порядком точности — четвертым, в то время как методы прямоугольников и трапеций — вторым.

Для анализа погрешности интегрирования с использованием сплайнов рассмотрим формулу (3.35). Первый член в ее правой части совпадает с правой частью формулы (3.27) для метода трапеций. Следовательно, второй член характеризует поправку к методу трапеций, которую дает использование сплайнов.

Как следует из формулы (3.32), коэффициенты c_i выражаются через вторые производные $\varphi_i''(x)$:

$$c_i = \frac{1}{2}\,\varphi_i''(x_{i-1}) \approx \frac{1}{2}\,y_{i-1}''.$$

Это позволяет оценить второй член правой части формулы (3.35):

$$\frac{h_i^3}{12}\,(c_i + c_{i+1}) \approx \frac{h_i^3}{12}\,y''_{i-1/2}.$$

Полученная оценка показывает (см. (3.40)), что добавка к формуле трапеций, которую дает использование сплайнов, компенсирует погрешность самой формулы трапеций. Можно показать, что метод сплайнов, как и метод Симпсона, имеет четвертый порядок точности.

Рассмотрев разные методы численного интегрирования, трудно сравнивать их достоинства и недостатки. Любая попытка такого сравнения непременно поставит перед нами альтернативный вопрос: что больше, h^2y'' или h^4y^{IV}? Все зависит от самой функции $y = f(x)$ и поведения ее производных.

Уточнение результатов численного интегрирования можно проводить по-разному. В частности, в представленном на рис. 3.4 алгоритме с использованием метода Симпсона проводится сравнение двух значений интеграла I_1 и I_2, полученных при разбиениях отрезка $[a, b]$ соответственно с шагами h и $h/2$. Аналогичный алгоритм можно построить и для других методов.

Здесь мы упомянем другую схему уточнения значения интеграла — *процесс Эйткена*. Он дает возможность оценить погрешность $O(h^r)$ метода и указывает алгоритм уточнения результатов. Расчет проводится последовательно три раза при различных шагах разбиения h_1, h_2, h_3, причем их отношения постоянны: $h_2/h_1 = h_3/h_2 = q$ (например, при делении шага пополам $q = 0.5$). Пусть в результате численного интегрирования получены значения интеграла I_1, I_2, I_3. Тогда уточненное значение интеграла вычисляется по формуле

$$I \approx I_1 - \frac{(I_1 - I_2)^2}{I_1 - 2I_2 + I_3}\,,$$

а порядок точности используемого метода численного интегрирования оценивается соотношением

$$r \approx \frac{1}{\ln q} \ln \frac{I_3 - I_2}{I_2 - I_1}\,.$$

Уточнение значения интеграла можно также проводить методом Рунге – Ромберга (см. § 5, п. 5).

6. Адаптивные алгоритмы. Из анализа погрешностей методов численного интегрирования следует, что точность получаемых результатов зависит как от характера изменения подынтегральной функции, так и от шага интегрирования. Будем считать, что величину шага мы задаем. При этом ясно, что для достижения сравнимой точности при интегрировании слабо меняющейся функции шаг можно выбирать большим, чем при интегрировании резко меняющихся функций.

На практике нередко встречаются случаи, когда подынтегральная функцияменяется по-разному на отдельных участках отрезка интегрирования. Это обстоятельство требует такой организации экономичных численных алгоритмов, при которой они автоматически приспосабливались бы к характеру изменения функции. Такие алгоритмы называются *адаптивными* (*приспосабливающимися*). Они позволяют вводить разные значения шага интегрирования на отдельных участках отрезка интегрирования. Это дает возможность уменьшить машинное время без потери точности результатов расчета. Подчеркнем, что этот подход используется обычно при задании подынтегральной функции $y = f(x)$ в виде формулы, а не в табличном виде.

Программа, реализующая адаптивный алгоритм численного интегрирования, входит обычно в виде стандартной подпрограммы в математическое обеспечение компьютера. Пользователь готовой программы задает границы отрезка интегрирования a, b, допустимую абсолютную погрешность ε и составляет блок программы для вычисления значения подынтегральной функции $f(x)$. Программа вычисляет значение интеграла I с заданной погрешностью ε, т. е.

$$\left| \int_a^b f(x)\,dx - I \right| \leqslant \varepsilon. \tag{3.43}$$

Разумеется, не для всякой функции можно получить результат с заданной погрешностью. Поэтому в программе может быть предусмотрено сообщение пользователю о недостижимости заданной погрешности. Интеграл при этом вычисляется с максимально возможной точностью, и программа выдает эту реальную точность.

Рассмотрим принцип работы адаптивного алгоритма. Первоначально отрезок $[a, b]$ разбиваем на n частей. В дальнейшем каждый такой элементарный отрезок делим последовательно пополам. Окончательное разбиение отрезка зависит от подынтегральной функции и допустимой погрешности ε.

К каждому элементарному отрезку $[x_{i-1}, x_i]$ применяем формулы численного интегрирования при двух различных его разбиениях. Получаем приближения $I_i^{(1)}$ и $I_i^{(2)}$ для интеграла по этому отрезку:

$$I_i = \int_{x_{i-1}}^{x_i} f(x)\,dx. \tag{3.44}$$

Полученные значения сравниваем и проводим оценку их погрешности. Если погрешность находится в допустимых границах, то одно из этих приближений принимается за значение интеграла по данному элементарному отрезку. В противном случае происходит дальнейшее деление отрезка и вычисление новых приближений. С целью экономии машинного

времени точки деления располагаются таким образом, чтобы использовались вычисленные значения функции в точках предыдущего разбиения.

Например, при вычислении интеграла (3.44) по формуле Симпсона отрезок $[x_{i-1}, x_i]$ сначала разбиваем на две части с шагом $h_i/2$ и вычисляем значение $I_i^{(1)}$. Потом вычисляем $I_i^{(2)}$ с шагом $h_i/4$, $I_i^{(3)}$ с шагом $h_i/8$ и т. д. Получим выражения

$$I_i^{(1)} = \frac{h_i}{6}\left[f(x_{i-1}) + 4f\left(x_{i-1} + \frac{h_i}{2}\right) + f(x_i)\right], \tag{3.45}$$

$$I_i^{(2)} = \frac{h_i}{12}\left[f(x_{i-1}) + 4f\left(x_{i-1} + \frac{h_i}{4}\right) + 2f\left(x_{i-1} + \frac{h_i}{2}\right) + 4f\left(x_{i-1} + \frac{3h_i}{4}\right) + f(x_i)\right], \tag{3.46}$$

$$I_i^{(3)} = \frac{h_i}{24}\left[f(x_{i-1}) + 4f\left(x_{i-1} + \frac{h_i}{8}\right) + 2f\left(x_{i-1} + \frac{h_i}{4}\right) + 4f\left(x_{i-1} + \frac{3h_i}{8}\right) + 2f\left(x_{i-1} + \frac{h_i}{2}\right) + 4f\left(x_{i-1} + \frac{5h_i}{8}\right) + 2f\left(x_{i-1} + \frac{3h_i}{4}\right) + 4f\left(x_{i-1} + \frac{7h_i}{8}\right) + f(x_i)\right], \tag{3.47}$$

. . . . . .

Процесс деления отрезка пополам и вычисления уточненных значений $I_i^{(k)}$ и $I_i^{(k+1)}$ $(k = 1, 2, \dots)$ продолжается до тех пор, пока их разность станет не больше некоторой заданной величины δ_i, зависящей от ε и h:

$$\left|I_i^{(k)} - I_i^{(k+1)}\right| \leqslant \delta_i. \tag{3.48}$$

Аналогичная процедура проводится для всех n элементарных отрезков. Величина $I = \sum\limits_{i=1}^{n} I_i$ принимается в качестве искомого значения интеграла. Условия (3.48) и соответствующий выбор величин δ_i обеспечивают выполнение условия (3.43).

Замечание. Нетрудно увидеть, что в формулах (3.45) – (3.47) значения функции $f(x)$ во внутренних точках отрезка $[x_{i-1}, x_i]$ первый раз появляются в сумме, заключенной в квадратные скобки, с коэффициентом 4. В формулах для последующих приближений те же слагаемые входят в сумму с коэффициентом 2. Этот факт позволяет организовать процесс последовательного деления отрезка $[x_{i-1}, x_i]$ пополам так, чтобы при нахождении приближения $I_i^{(k+1)}$ значения функции $f(x)$, вычисленные ранее при нахождении предыдущих приближений, заново не вычислялись. Например, можно отдельно запоминать сумму значений функции, вычисленных на текущем шаге, сумму значений функции, вычисленных на предыдущем шаге, а также сумму прочих слагаемых, стоящих в квадратных скобках.

7. О других методах. Особые случаи. Кроме рассмотренных выше методов численного интегрирования существует ряд других. Дадим краткий обзор некоторых из них.

Формулы Ньютона – Котеса получаются путем замены подынтегральной функции интерполяционным многочленом Лагранжа с разбиением отрезка интегрирования на n равных частей. Получающиеся формулы используют значения подынтегральной функции в узлах интерполяции и являются точными для всех многочленов некоторой степени [1)], зависящей от числа узлов. Точность формул растет с увеличением степени интерполяционного многочлена [2)].

Метод Гаусса не предполагает разбиения отрезка интегрирования на равные промежутки. Формулы численного интегрирования интерполяционного типа ищутся такими, чтобы они обладали наивысшим порядком точности при заданном числе узлов. Узлы и коэффициенты формул численного интегрирования находятся из условий обращения в нуль их остаточных членов для всех многочленов максимально высокой степени.

Формула Эрмита, являющаяся частным случаем формул Гаусса, использует многочлены Чебышева для вычисления интегралов вида

$$\int\limits_{-1}^{1} \frac{f(x)\,dx}{\sqrt{1-x^2}}\,.$$

Получающаяся формула характерна тем, что все коэффициенты α_i в (3.23) равны.

Метод Маркова состоит в том, что при выводе формул Гаусса вводятся дополнительные предположения о совпадении точек разбиения отрезка по крайней мере с одним из его концов.

Формула Чебышева представляет интеграл в виде

$$\int\limits_{-1}^{1} f(x)\,dx \approx k\sum_{i=0}^{n} f(x_i). \tag{3.49}$$

При этом решается следующая задача:

найти точки x_0, x_1, ... , x_n и коэффициент k такие, при которых формула (3.49) точна для всех многочленов как можно большей степени.

Формула Эйлера использует не только значения подынтегральной функции в точках разбиения, но и значения ее производных до некоторого порядка на границах отрезка.

[1)] Это означает, что если подынтегральная функция — многочлен, то квадратурная формула дает точное значение интеграла (естественно, без учета погрешностей округления).

[2)] Заметим, что формулы прямоугольников, трапеций и Симпсона являются частными случаями формул Ньютона – Котеса.

Рассмотрим *особые случаи численного интегрирования*:
а) подынтегральная функция разрывна на отрезке интегрирования;
б) несобственные интегралы.

а) В ряде случаев подынтегральная функция $f(x)$ или ее производные в некоторых внутренних точках c_k $(k = 1, 2, \dots)$ отрезка интегрирования $[a, b]$ терпят разрыв. В этом случае интеграл вычисляют численно для каждого участка непрерывности и результаты складывают. Например, в случае одной точки разрыва $x = c$ $(a < c < b)$ имеем

$$\int_a^b f(x)\,dx = \int_a^c f(x)\,dx = \int_c^b f(x)\,dx.$$

Для вычисления каждого из стоящих в правой части интегралов можно использовать рассмотренные выше методы.

б) Не так просто обстоит дело с вычислением *несобственных интегралов*. Напомним, что к такому типу относятся интегралы, которые имеют хотя бы одну бесконечную границу интегрирования или подынтегральную функцию, обращающуюся в бесконечность хотя бы в одной точке отрезка интегрирования.

Рассмотрим сначала *интеграл с бесконечной границей интегрирования*, например интеграл вида

$$\int_a^{+\infty} f(x)\,dx, \quad 0 < a < +\infty.$$

Существует несколько приемов вычисления таких интегралов.

Можно попытаться ввести замену переменных $x = a/(1 - t)$, которая превращает промежуток интегрирования $[0, +\infty)$ в отрезок $[0, 1]$. При этом подынтегральная функция и первые ее производные до некоторого порядка должны оставаться ограниченными.

Еще один прием состоит в том, что бесконечная граница заменяется некоторым достаточно большим числом A так, чтобы принятое значение интеграла отличалось от исходного на некоторый малый остаток, т. е.

$$\int_a^{+\infty} f(x)\,dx = \int_a^A f(x)\,dx + R, \quad R = \int_A^{+\infty} f(x)\,dx. \tag{3.50}$$

Если функция обращается в бесконечность в некоторой точке $x = c$ конечного отрезка интегрирования, то можно попытаться выделить особенность, представив подынтегральную функцию в виде суммы двух функций: $f(x) = \varphi(x) + \psi(x)$. При этом $\varphi(x)$ ограничена, а $\psi(x)$ имеет особенность в данной точке, но интеграл (несобственный) от нее может быть вычислен аналитически. Тогда численный метод используется только для интегрирования ограниченной функции $\varphi(x)$. Можно также использовать

представление интеграла, аналогичное (3.50):

$$\int_a^b f(x)\,dx = \int_a^{c_1} f(x)\,dx + \int_{c_1}^b f(x)\,dx + R, \quad R = \int_{c_1}^c f(x)\,dx + \int_c^{c_2} f(x)\,dx.$$

Здесь c_1 и c_2 — некоторые близкие к c числа.

Еще один вид несобственных интегралов (сингулярные интегралы), имеющий важное прикладное значение, будет рассмотрен в дальнейшем в разделе, посвященном сингулярным интегральным уравнениям (см. гл. 9, § 3).

8. Кратные интегралы. Численные методы используются также для вычисления кратных интегралов. Ограничимся здесь рассмотрением *двойных интегралов* вида

$$\iint\limits_G f(x,y)\,dx\,dy. \tag{3.51}$$

Одним из простейших способов вычисления этого интеграла является *метод ячеек*. Рассмотрим сначала случай, когда областью интегрирования G является прямоугольник: $a \leqslant x \leqslant b, c \leqslant y \leqslant d$. По теореме о среднем найдем среднее значение функции $f(x,y)$:

$$\bar{f}(x,y) = \frac{1}{S}\iint\limits_G f(x,y)\,dx\,dy, \quad S = (b-a)(d-c). \tag{3.52}$$

Будем считать, что среднее значение приближенно равно значению функции в центре прямоугольника, т. е. $\bar{f}(x,y) = f(\bar{x},\bar{y})$. Тогда из (3.52) получим выражение для приближенного вычисления двойного интеграла:

$$\begin{gathered}\iint\limits_G f(x,y)\,dx\,dy \approx Sf(\bar{x},\bar{y}),\\ \bar{x} = \frac{a+b}{2}, \quad \bar{y} = \frac{c+d}{2}.\end{gathered} \tag{3.53}$$

Точность этой формулы можно повысить, если разбить область G на прямоугольные ячейки ΔG_{ij} (рис. 3.5): $x_{i-1} \leqslant x \leqslant x_i$ $(i = 1, 2, \ldots, M)$, $y_{j-1} \leqslant y \leqslant y_j$ $(j = 1, 2, \ldots, N)$. Применяя к каждой ячейке формулу (3.53), получаем

$$\iint\limits_{\Delta G_{ij}} f(x,y)\,dx\,dy \approx f(\bar{x}_i,\bar{y}_j)\Delta x_i \Delta y_j.$$

Суммируя эти выражения по всем ячейкам, находим значение двойного интеграла:

$$\iint\limits_G f(x,y)\,dx\,dy \approx \sum_{i=1}^{M}\sum_{j=1}^{N} f(\bar{x}_i,\bar{y}_j)\Delta x_i \Delta y_j. \tag{3.54}$$

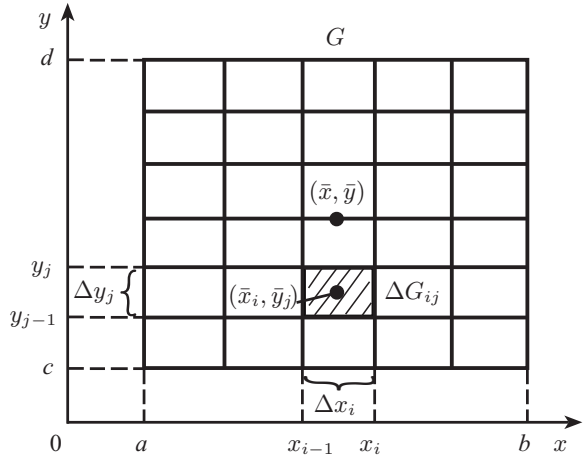


Рис. 3.5

В правой части стоит интегральная сумма; поэтому при неограниченном уменьшении периметров ячеек (или стягивании их в точки) эта сумма стремится к значению интеграла для любой непрерывной функции $f(x, y)$.

Можно показать, что погрешность такого приближения интеграла для одной ячейки оценивается соотношением

$$R_{ij} \approx \frac{\Delta x_i \Delta y_j}{24} \left[\left(\frac{b-a}{M} \right)^2 f''_{xx} + \left(\frac{d-c}{N} \right)^2 f''_{yy} \right].$$

Суммируя эти выражения по всем ячейкам и считая все их площади одинаковыми, получаем оценку погрешности метода ячеек в виде

$$R = O(1/M^2 + 1/N^2) = O(\Delta x^2 + \Delta y^2).$$

Таким образом, формула (3.54) имеет второй порядок точности. Для уменьшения погрешности можно использовать обычные *методы сгущения узлов сетки*. При этом по каждой переменной шаги уменьшают в одинаковое число раз, т. е. отношение M/N остается постоянным.

Если область G непрямоугольная, то в ряде случаев ее целесообразно привести к прямоугольному виду путем соответствующей *замены переменных*. Например, пусть область задана в виде криволинейного четырехугольника: $a \leqslant x \leqslant b$, $\varphi_1(x) \leqslant y \leqslant \varphi_2(x)$. Данную область можно привести к прямоугольному виду с помощью замены

$$t = \frac{y - \varphi_1(x)}{\varphi_2(x) - \varphi_1(x)}, \quad 0 \leqslant t \leqslant 1.$$

Кроме того, формула (3.54) может быть обобщена и на случай более сложных областей.

Другим довольно распространенным методом вычисления кратных интегралов является их *сведение к последовательному вычислению определенных интегралов*. Интеграл (3.51) для прямоугольной области можно записать в виде

$$\iint\limits_G f(x,y)\,dx\,dy = \int\limits_a^b F(x)\,dx, \quad F(x) = \int\limits_c^d f(x,y)\,dy.$$

Для вычисления обоих определенных интегралов могут быть использованы рассмотренные ранее численные методы.

Если область G имеет более сложную структуру, то она либо приводится к прямоугольному виду с помощью замены переменных, либо разбивается на простые элементы.

Для вычисления кратных интегралов используется также *метод замены подынтегральной функции многомерным интерполяционным многочленом*. Вычисление коэффициентов этих многочленов для простых областей обычно не вызывает затруднений.

Существует ряд других численных методов вычисления кратных интегралов. Среди них особое место занимает метод статистических испытаний, который мы вкратце изложим.

9. Метод Монте-Карло. Во многих задачах исходные данные носят случайный характер, поэтому для их решения должен применяться статистико-вероятностный подход. На основе таких подходов построен ряд численных методов, которые учитывают случайный характер вычисляемых или измеряемых величин. К ним принадлежит и *метод статистических испытаний*, называемый также *методом Монте-Карло* [1)], который применяется к решению некоторых задач вычислительной математики, в том числе и для вычисления интегралов.

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

$$\mathsf{M}\xi = x.$$

Проводится серия n независимых испытаний, в результате которых получается (генерируется) последовательность n случайных чисел $\xi_1, \xi_2, \ldots$ $\ldots, \xi_n$ (выборка), имеющих то же распределение, что и ξ, и по совокупности этих значений находится выборочное среднее $\bar{\xi}$, которое является статистической оценкой $\mathsf{M}\xi$. Искомая величина x полагается приближенно равной этой оценке

$$x \approx \bar{\xi} = \frac{1}{n}(\xi_1 + \xi_2 + \ldots + \xi_n).$$

[1)] Название метода произошло от названия города Монте-Карло, знаменитого своими казино, в которых играют в рулетку, являющуюся одним из простейших генераторов случайных чисел.

Пусть η — равномерно распределенная па отрезке $[0,1]$ случайная величина. Это означает, что ее плотность распределения задается соотношением

$$p_\eta(x) = \begin{cases} 0, & x < 0, \\ 1, & 0 \leqslant x \leqslant 1, \\ 0, & x > 1. \end{cases}$$

Тогда любая функция $\xi = f(\eta)$ также будет случайной величиной, и ее математическое ожидание равно

$$\mathsf{M}\xi = \int\limits_{-\infty}^{+\infty} f(x)p_\eta(x)\,dx = \int\limits_0^1 f(x)\,dx.$$

Следовательно, читая это равенство в обратном порядке, приходим к выводу, что интеграл $\int\limits_0^1 f(x)\,dx$ может быть вычислен как оценка математического ожидания некоторой случайной величины ξ, которая является функцией случайной величины η с равномерным законом распределения, причем оценка $\mathsf{M}\xi$ определяется независимыми реализациями η_i случайной величины η:

$$\int\limits_0^1 f(x)\,dx \approx \bar{\xi} = \frac{1}{n}\sum_{i=1}^{n} f(\eta_i).$$

Аналогично могут быть вычислены и кратные интегралы. Для двойного интеграла получим

$$\iint\limits_G f(x,y)\,dx\,dy \approx \frac{1}{n}\sum_{i=1}^{n} f(\eta_i,\zeta_i),$$

где G — квадрат $0 \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1$; $\eta_i,\ \zeta_i$ — независимые реализации случайных величин $\eta,\ \zeta,$ равномерно распределенных на отрезке $[0,1]$.

Для использования метода Монте-Карло при вычислении определенных интегралов, как и в других его приложениях, необходимо вырабатывать последовательности случайных чисел с заданным законом распределения. Существуют различные способы генерирования таких чисел.

Можно построить некоторый физический процесс (генератор) для выработки случайных величин, однако при использовании компьютера этот способ не применяется, поскольку, во-первых, трудно дважды получить одинаковые совокупности случайных чисел, которые необходимы при отладке программ, а во-вторых, такой физический генератор существенно усложнил бы конструкцию компьютера.

Известны многие таблицы случайных чисел, которые вычислялись независимо. Их можно ввести в компьютер и при необходимости обращаться к ним.

В настоящее время наиболее распространенный способ выработки случайных чисел на компьютере состоит в том, что в памяти хранится некоторый алгоритм получения таких чисел по мере потребности в них (подобно тому как вычисляются значения элементарных функций, а не хранятся их таблицы). Поскольку эти числа генерируются по наперед заданному алгоритму, то они не совсем случайны (*псевдослучайны*), хотя и обладают свойственными случайным числам статистическими характеристиками. В современных языках программирования такие алгоритмы реализованы в виде подпрограмм — датчиков случайных чисел.

Упражнения

1. Функция $y = f(x)$ задана в табличной форме:

x	0	0.2	0.4	0.6	0.8	1.0
y	1.24	1.03	1.36	1.85	2.43	3.14

Вычислить:
а) значения производной в точках $x = 0,\ 0.4,\ 1.0$ с первым и вторым порядками точности;
б) вторую производную в этих же точках со вторым в третьим порядками точности.

2. Записать алгоритм вычисления производной функции, заданной таблицей с постоянным шагом на некотором отрезке.

3*. Получить формулу Ромберга (3.17).
У к а з а н и е. Воспользоваться представлением погрешности в виде

$$R = h^p\varphi_1(x) + h^{p+1}\varphi_2(x) + \ldots + h^{p+q-2}\varphi_{p+q-1}(x) + O(h^{p+q-1}).$$

4. Для функции $f(x, y) = \sin(x + y^2)$ вычислить все частные производные до второго порядка включительно в точке $(0, 0)$, используя различные аппроксимации, приведенные в п. 6 § 1. Принять $h_1 = h_2 = 0.1$. Сравнить полученные результаты с точными значениями производных.

5. Вычислить $\int\limits_0^1 e^{x^2}\,dx$, используя методы прямоугольников, трапеций и Симпсона. Отрезок интегрирования разделить на десять равных частей.

6. Используя процесс Эйткена и метод трапеций, вычислить $\int\limits_1^2 \frac{1}{x}\sin\frac{\pi x}{2}\,dx$. Число шагов интегрирования принять равным 4, 8, 46.

7. Записать алгоритм решения упр. 9.

8. Записать адаптивный алгоритм вычисления интеграла по методу Симпсона, воспользовавшись замечанием, сделанным в п. 6 § 2.

9. С помощью метода Монте-Карло вычислить площадь фигуры, заданной уравнением $\sqrt[3]{x^2} + \sqrt[3]{y^2} = 1$. Принять n равным 10^4, 10^5, 10^6. Сравнить ответы с точным значением площади.

ГЛАВА 4

СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

§ 1. Основные понятия

1. Линейные системы. К решению систем линейных уравнений сводятся многочисленные практические задачи. Можно с полным основанием утверждать, что решение линейных систем является одной из самых распространенных и важных задач вычислительной математики.

Запишем *систему n линейных алгебраических уравнений с n неизвестными*:

$$\begin{aligned} a_{11}x_1 &+ a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 &+ a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 &+ a_{n2}x_2 + \ldots + a_{nn}x_n = b_n. \end{aligned} \tag{4.1}$$

Совокупность коэффициентов этой системы запишем в виде таблицы:

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}.$$

Данная таблица n^2 элементов, состоящая из n строк и n столбцов, называется *квадратной матрицей* порядка n. Если подобная таблица содержит mn элементов, расположенных в m строках и n столбцах, то она называется *прямоугольной матрицей*.

Используя понятие матрицы A, систему уравнений (4.1) можно записать в векторно-матричном виде:

$$A\mathbf{x} = \mathbf{b}, \tag{4.2}$$

где $\mathbf{x}$ и $\mathbf{b}$ — вектор-столбец неизвестных и вектор-столбец правых частей соответственно:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix},$$

или, в более компактной записи,

$$\mathbf{x} = \{x_1, x_2, \ldots, x_n\}, \quad \mathbf{b} = \{b_1, b_2, \ldots, b_n\}.$$

В ряде случаев получаются системы уравнений с некоторыми специальными видами матриц. Вот некоторые примеры таких матриц:

$$A=\begin{pmatrix} 2 & 1 & -1\\ 1 & 3 & 2\\ -1 & 2 & 4\end{pmatrix},\qquad B=\begin{pmatrix} 1 & 2 & 3\\ 0 & -1 & 1\\ 0 & 0 & 2\end{pmatrix},$$

$$C=\begin{pmatrix} 1 & 2 & 1 & 0 & 0 & 0\\ 2 & -1 & 2 & 0 & 0 & 0\\ 3 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 4 & -1 & 1\\ 0 & 0 & 0 & -1 & 4 & -1\\ 0 & 0 & 0 & 2 & 1 & 1\end{pmatrix},\qquad F=\begin{pmatrix} 3 & 2 & 0 & 0 & 0 & 0\\ 1 & 2 & -1 & 0 & 0 & 0\\ 0 & 3 & -2 & 2 & 0 & 0\\ 0 & 0 & 1 & 2 & -1 & 0\\ 0 & 0 & 0 & 1 & 3 & 1\\ 0 & 0 & 0 & 0 & -1 & 3\end{pmatrix},$$

$$E=\begin{pmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix},\qquad O=\begin{pmatrix} 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{pmatrix}.$$

Здесь:

A — симметрическая матрица (ее элементы расположены симметрично относительно главной диагонали ($a_{ij}=a_{ji}$));

B — верхняя треугольная матрица с равными нулю элементами, расположенными ниже диагонали;

C — клеточная матрица (ее ненулевые элементы составляют отдельные группы (клетки));

F — ленточная матрица (ее ненулевые элементы составляют «ленту», параллельную диагонали (в данном случае ленточная матрица F одновременно является также трехдиагональной));

E — единичная матрица (частный случай диагональной);

O — нулевая матрица.

Определителем (*детерминантом*) матрицы A n-го порядка называется число D, равное

$$D=\det A=\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n}\\ a_{21} & a_{22} & \dots & a_{2n}\\ \dots & \dots & \dots & \dots\\ a_{n1} & a_{n2} & \dots & a_{nn}\end{vmatrix}=\sum(-1)^k a_{1\alpha}a_{2\beta}\dots a_{n\omega}. \qquad (4.3)$$

Здесь индексы α, β, $\dots$, ω пробегают все возможные $n!$ перестановок номеров $1, 2, \dots, n$; k — число инверсий в данной перестановке [1)].

Необходимым и достаточным условием существования единственного решения системы линейных уравнений является условие $D\neq 0$. В случае равенства нулю определителя системы матрица называется *вырожденной*; при этом система линейных уравнений (4.1) либо не имеет решения, либо имеет их бесконечное множество.

[1)] Под инверсией понимается обмен двух индексов местами; с помощью таких обменов перестановка $\alpha, \beta, \dots, \omega$ получается из перестановки $1, 2, \dots, n$.

Все эти случаи легко проиллюстрировать геометрически для системы

$$\begin{aligned} a_1x + b_1y &= c_1, \\ a_2x + b_2y &= c_2. \end{aligned} \tag{4.4}$$

Каждое уравнение описывает прямую на плоскости; координаты точки пересечения указанных прямых являются решением системы (4.4).

Рассмотрим три возможных случая взаимного расположения двух прямых на плоскости:

1) прямые пересекаются, т. е. коэффициенты системы (4.4) не пропорциональны:

$$\frac{a_1}{a_2} \neq \frac{b_1}{b_2}\,; \tag{4.5}$$

2) прямые параллельны , т. е. коэффициенты системы (4.4) подчиняются условиям

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}\,; \tag{4.6}$$

3) прямые совпадают, т. е. все коэффициенты (4.4) пропорциональны:

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}\,. \tag{4.7}$$

Запишем определитель D системы (4.4) в виде

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}.$$

Отметим, что при выполнении условия (4.5) $D \neq 0$, и система (4.4) имеет единственное решение. В случаях отсутствия решения или при бесконечном множестве решений имеют место соответственно соотношения (4.6) или (4.7), из которых получаем $D = 0$.

На практике, особенно при вычислениях на компьютере, когда происходят округление или отбрасывание младших разрядов чисел, далеко не всегда удается получить точное равенство определителя нулю. При $D \approx 0$ прямые могут оказаться почти параллельными (в случае системы двух уравнений); координаты точки пересечения этих прямых весьма чувствительны к изменению коэффициентов системы (см. рис. 4.1).

Таким образом, малые погрешности вычислений или исходных данных могут привести к существенным погрешностям в решении. Такие системы уравнений называются *плохо обусловленными*.

Заметим, что условие $D \approx 0$ является необходимым для плохой обусловленности системы линейных уравнений, но не достаточным. Например, система уравнений n-го порядка с диагольной матрицей с элементами $a_{ii} = 0.1$ не является плохо обусловленной, хотя ее определитель мал ($D = 10^{-n}$). Строгий критерий плохой обусловленности системы линейных уравнений можно найти в более полных курсах по численным методам.

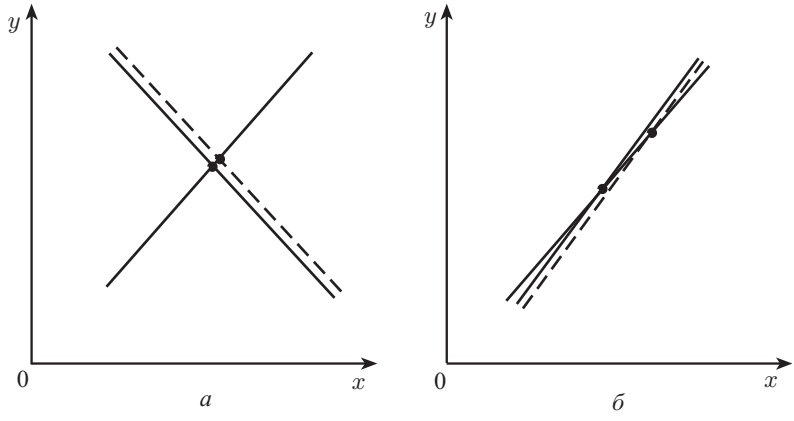


Рис. 4.1. Геометрическая иллюстрация системы двух уравнений: при малом изменении параметров одной из прямых координаты точки пересечения мало изменяются в случае *а* и заметно изменяются в случае *б*

Приведенные соображения справедливы и для любого числа уравнений системы (4.1), хотя в случае $n > 3$ нельзя привести простые геометрические иллюстрации. При $n = 3$ каждое уравнение описывает плоскость в пространстве, и в случае почти параллельных плоскостей или линий их попарного пересечения получаем плохо обусловленную систему трех уравнений.

2. О методах решения линейных систем. Методы решения систем линейных уравнений делятся на две группы — прямые и итерационные. *Прямые методы* используют конечные соотношения (формулы) для вычисления неизвестных. Они дают решение после выполнения заранее известного числа операций. Эти методы сравнительно просты и наиболее универсальны, т. е. пригодны для решения широкого класса линейных систем.

Вместе с тем прямые методы имеют и ряд недостатков. Как правило, они требуют хранения в оперативной памяти компьютера сразу всей матрицы, и при больших значениях n расходуется много места в памяти. Далее, прямые методы обычно не учитывают структуру матрицы при большом числе нулевых элементов в разреженных матрицах (например, клеточных или ленточных) эти элементы занимают место в памяти машины, и над ними проводятся арифметические действия. Исключением здесь является метод прогонки (см. § 2, п. 4). Существенным недостатком прямых методов является также накапливание погрешностей в процессе решения, поскольку вычисления на любом этапе используют результаты предыдущих операций. Это особенно опасно для больших систем, когда резко возрастает общее число операций, а также для плохо обусловленных систем, весьма чувствительных к погрешностям. В связи с этим прямые методы используются обычно для не слишком больших ($n \lesssim 1000$) систем с плотно заполненной матрицей и не близким к нулю определителем.

Отметим еще, что прямые методы решения линейных систем иногда называют *точными*, поскольку решение выражается в виде точных формул через коэффициенты системы. Однако точное решение может быть получено лишь при точном выполнении вычислений (и, разумеется, при точных значениях коэффициентов системы). На практике же при использовании компьютеров вычисления проводятся с погрешностями. Поэтому неизбежны погрешности и в окончательных результатах.

Итерационные методы — это методы последовательных приближений. В них необходимо задать некоторое приближенное решение — *начальное приближение*. После этого с помощью некоторого алгоритма проводится один цикл вычислений, называемый *итерацией*. В результате итерации находят новое приближение. Итерации проводятся до получения решения с требуемой точностью. Алгоритмы решения линейных систем с использованием итерационных методов обычно более сложные по сравнению с прямыми методами. Объем вычислений заранее определить трудно.

Тем не менее итерационные методы в ряде случаев предпочтительнее. Они требуют хранения в памяти машины не всей матрицы системы, а лишь нескольких векторов с n компонентами. Иногда элементы матрицы можно совсем не хранить, а вычислять их по мере необходимости. Погрешности окончательных результатов при использовании итерационных методов не накапливаются, поскольку точность вычислений в каждой итерации определяется лишь результатами предыдущей итерации и практически не зависит от ранее выполненных вычислений. Эти достоинства итерационных методов делают их особенно полезными в случае большого числа уравнений, а также плохо обусловленных систем. Следует отметить, что при этом сходимость итераций может быть очень медленной; поэтому ищутся эффективные пути ее ускорения.

Итерационные методы могут использоваться для уточнения решений, полученных с помощью прямых методов. Такие смешанные алгоритмы обычно довольно эффективны, особенно для плохо обусловленных систем. В последнем случае могут также применяться методы регуляризации.

3. Другие задачи линейной алгебры. Кроме решения систем линейных уравнений существуют другие задачи линейной алгебры — вычисление определителя, обратной матрицы, собственных значений матрицы и др.

Легко вычисляются лишь определители невысоких порядков и некоторые специальные типы определителей. В частности, для определителей второго и третьего порядков соответственно имеем

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21},$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{21}a_{32}a_{13} -$$

$$- a_{31}a_{22}a_{13} - a_{21}a_{12}a_{33} - a_{32}a_{23}a_{11}.$$

Определитель треугольной матрицы равен произведению ее элементов, расположенных на главной диагонали: $D = a_{11}a_{22}\dots a_{nn}$. Отсюда также следует, что определитель единичной матрицы равен единице, а нулевой — нулю: $\det E = 1,\ \det O = 0$.

В общем случае вычисление определителя оказывается значительно более трудоемким. Определитель D порядка n имеет вид (4.3)

$$D = \sum (-1)^k a_{1\alpha}a_{2\beta}\dots a_{n\omega}.$$

Из этого выражения следует, что определитель равен сумме $n!$ слагаемых, каждое из которых является произведением n элементов. Поэтому для вычисления определителя порядка n (без использования специальных приемов) требуется $(n-1)n!$ умножений и $n! - 1$ сложений, т. е. общее число арифметических операций равно

$$N = n \cdot n! - 1 \approx n \cdot n!. \tag{4.8}$$

Оценим значения N в зависимости от порядка n определителя:

n	3	10	20
N	17	$3.6 \cdot 10^7$	$5 \cdot 10^{19}$

Можно подсчитать время вычисления таких определителей на компьютере с заданным быстродействием. Примем для определенности среднее быстродействие равным 10 млн. операций в секунду. Тогда для вычисления определителя 10-го порядка потребуется около 3.6 с, а при $n = 20$ — свыше 150 тыс. лет.

Приведенные оценки указывают на необходимость разработки и использования экономичных численных методов, позволяющих эффективно проводить вычисления определителей. В § 2 будет рассмотрен один из таких методов.

Матрица A^{-1} называется *обратной* по отношению к квадратной матрице A, если их произведение равно единичной матрице: $AA^{-1} = A^{-1}A = E$. В линейной алгебре доказывается, что всякая невырожденная матрица A (т. е. с отличным от нуля определителем D) имеет обратную. При этом

$$\det A^{-1} = 1/D.$$

Запишем исходную матрицу в виде

$$A = \begin{pmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1} & \dots & a_{ij} & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{nj} & \dots & a_{nn} \end{pmatrix}.$$

Минором элемента a_{ij} называется определитель $(n-1)$-го порядка, образованный из определителя матрицы A зачеркиванием i-й строки и j-го столбца.

Алгебраическим дополнением A_{ij} элемента a_{ij} называется его минор, взятый со знаком плюс, если сумма $i+j$ номеров строки i и столбца j четная, и со знаком минус, если эта сумма нечетная, т. е.

$$A_{ij} = (-1)^{i+j} \begin{vmatrix} a_{11} & \dots & a_{1,j-1} & a_{1,j+1} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i-1,1} & \dots & a_{i-1,j-1} & a_{i-1,j+1} & \dots & a_{i-1,n} \\ a_{i+1,1} & \dots & a_{i+1,j-1} & a_{i+1,j+1} & \dots & a_{i+1,n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{n,j-1} & a_{n,j+1} & \dots & a_{nn} \end{vmatrix}.$$

Каждый элемент z_{ij} $(i, j = 1, \dots, n)$ обратной матрицы $Z = A^{-1}$ равен отношению алгебраического дополнения A_{ji} элемента a_{ji} (не a_{ij}) исходной матрицы A к значению ее определителя D:

$$Z = A^{-1} = \begin{pmatrix} \dfrac{A_{11}}{D} & \dfrac{A_{21}}{D} & \dots & \dfrac{A_{n1}}{D} \\ \dfrac{A_{12}}{D} & \dfrac{A_{22}}{D} & \dots & \dfrac{A_{n2}}{D} \\ \dots & \dots & \dots & \dots \\ \dfrac{A_{1n}}{D} & \dfrac{A_{2n}}{D} & \dots & \dfrac{A_{nn}}{D} \end{pmatrix}.$$

Здесь, как и выше, можно также подсчитать число операций, необходимое для вычисления обратной матрицы без использования специальных методов. Это число равно сумме числа операций, с помощью которых вычисляются n^2 алгебраических дополнений, каждое из которых является определителем $(n-1)$-го порядка, и n^2 делений алгебраических дополнений на определитель D. Таким образом, общее число операций для вычислений обратной матрицы равно

$$N = [(n-1)\cdot(n-1)!-1]n^2 + n^2 + n\cdot n! - 1 = n^2\cdot n! - 1.$$

Как и для задачи вычисления определителя, полученное значение N указывает на необходимость применения эффективных методов обращения матриц.

Важной задачей линейной алгебры является также вычисление собственных значений матрицы. Этому вопросу будет посвящен § 4.

§ 2. Прямые методы

1. Вводные замечания. Одним из способов решения системы линейных уравнений является *правило Крамера*, согласно которому каждое неизвестное представляется в виде отношения определителей. Запишем его для системы

$$a_1 x + b_1 y = c_1,$$
$$a_2 x + b_2 y = c_2.$$

Тогда

$$x = D_1/D, \quad y = D_2/D,$$

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}, \quad D_1 = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}, \quad D_2 = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}.$$

Можно попытаться использовать это правило для решения систем уравнений произвольного порядка. Однако при большом числе уравнений потребуется выполнить огромное число арифметических операций, поскольку для вычислений n неизвестных необходимо найти значения определителей, число которых $n + 1$. Количество арифметических операций можно оценить с учетом формулы (4.8). При этом предполагаем, что определители вычисляются непосредственно — без использования экономичных методов. Тогда получим

$$N = (n + 1)(n \cdot n! - 1) + n.$$

Поэтому правило Крамера можно использовать лишь для решения систем, состоящих из нескольких уравнений.

Известен также метод решения линейной системы с использованием обратной матрицы. Система записывается в виде $A\mathbf{x} = \mathbf{b}$ (см. (4.2)). Тогда, умножая обе части этого векторного уравнения слева на обратную матрицу A^{-1}, получаем $\mathbf{x} = A^{-1}\mathbf{b}$. Однако если не использовать экономичных схем для вычисления обратной матрицы, этот способ также непригоден для практического решения линейных систем при больших значениях n из-за большого объема вычислений.

Наиболее распространенными среди прямых методов являются *метод исключения Гаусса* и его модификации. Ниже рассматривается применение метода исключения для решения систем линейных уравнений, а также для вычисления определителя и нахождения обратной матрицы.

2. Метод Гаусса. Он основан на приведении матрицы системы к треугольному виду. Это достигается последовательным исключением неизвестных из уравнений системы. Сначала с помощью первого уравнения исключается x_1 из всех последующих уравнений системы. Затем с помощью второго уравнения исключается x_2 из третьего и всех последующих уравнений. Этот процесс, называемый *прямым ходом метода Гаусса*, продолжается до тех пор, пока в левой части последнего (n-го) уравнения не останется лишь один член с неизвестным x_n, т. е. матрица системы будет приведена к треугольному виду.

Обратный ход метода Гаусса состоит в последовательном вычислении искомых неизвестных: решая последнее уравнение, находим единственное в этом уравнении неизвестное x_n. Далее, используя это значение, из предыдущего уравнения вычисляем x_{n-1} и т. д. Последним найдем x_1 из первого уравнения.

Заметим, что описанные процедуры применимы лишь для систем с невырожденной матрицей. В противном случае (при условии, что вычисления проводятся точно) с помощью метода Гаусса можно ответить на вопрос, имеет ли система бесконечное множество решений или не имеет ни одного. Однако эти случаи мы в дальнейшем рассматривать не будем, предполагая, что матрица системы невырожденная.

Рассмотрим применение метода Гаусса для системы

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \tag{4.9}$$

Для исключения x_1 из второго уравнения прибавим к нему первое, умноженное на $-a_{21}/a_{11}$. Затем, умножив первое уравнение на $-a_{31}/a_{11}$ и прибавив результат к третьему уравнению, также исключим из него x_1. Получим равносильную (4.9) систему уравнений вида

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a'_{22}x_2 + a'_{23}x_3 &= b'_2, \\ a'_{32}x_2 + a'_{33}x_3 &= b'_3; \end{aligned} \tag{4.10}$$

$$a'_{ij} = a_{ij} - \frac{a_{i1}}{a_{11}}\, a_{1j}, \quad i,j = 2,3,$$

$$b'_i = b_i - \frac{a_{i1}}{a_{11}}\, b_1, \qquad i = 2,3.$$

Теперь из третьего уравнения системы (4.10) нужно исключить x_2. Для этого умножим второе уравнение на $-a'_{32}/a'_{22}$ и прибавим результат к третьему. Получим

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a'_{22}x_2 + a'_{23}x_3 &= b'_2, \\ a''_{33}x_3 &= b''_3; \end{aligned} \tag{4.11}$$

$$a''_{33} = a'_{33} - \frac{a'_{32}}{a'_{22}}\, a'_{23}, \quad b''_3 = b'_3 - \frac{a'_{32}}{a'_{22}}\, b'_2.$$

Матрица системы (4.11) имеет треугольный вид. На этом заканчивается прямой ход метода Гаусса.

Заметим, что в процессе исключения неизвестных приходится выполнять операции деления на коэффициенты a_{11}, a'_{22} и т. д. Поэтому они должны быть отличны от нуля. В противном случае необходимо соответственным образом переставить уравнения системы. Перестановка уравнений должна быть предусмотрена в вычислительном алгоритме при его реализации на компьютере.

Обратный ход начинается с решения третьего уравнения системы (4.11):

$$x_3 = b_3''/a_3''.$$

Используя это значение, можно найти x_2 из второго уравнения, а затем x_1 из первого:

$$x_2 = \frac{1}{a_{22}'}\,(b_2' - a_{23}' x_3),$$

$$x_1 = \frac{1}{a_{11}}\,(b_1 - a_{12}x_2 - a_{13}x_3).$$

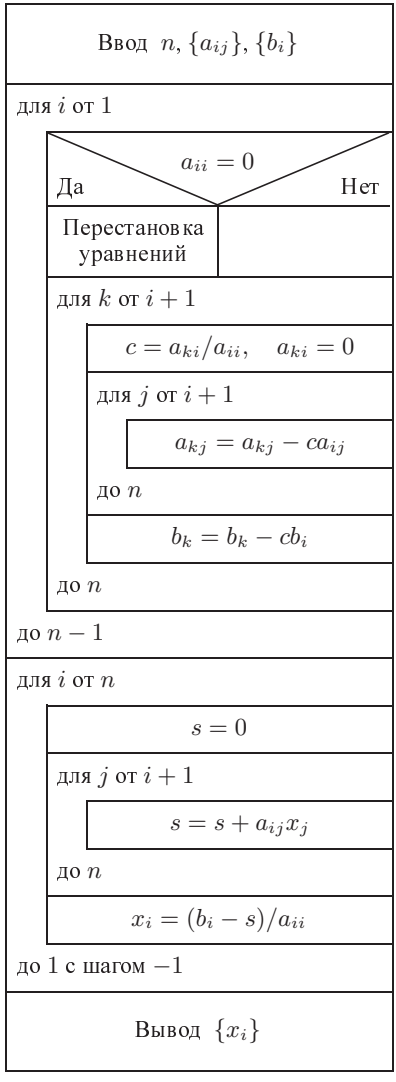


Рис. 4.2. Метод Гаусса

Аналогично строится вычислительный алгоритм для линейной системы с произвольным числом уравнений. На рис. 4.2 приведен алгоритм решения методом Гаусса системы n линейных уравнений вида (4.1). Он состоит из ввода исходных данных, двух циклов с переменной цикла i и вывода результатов. Первый цикл с переменной цикла i реализует прямой ход, а второй — обратный ход метода. Поясним смысл индексов: i — номер неизвестного, которое исключается из оставшихся $n - i$ уравнений при прямом ходе (а также номер того уравнения, с помощью которого исключается x_i) и номер неизвестного, которое определяется из i-го уравнения при обратном ходе; k — номер уравнения, из которого исключается неизвестное x_i при прямом ходе; j — номер столбца при прямом ходе и номер уже найденного неизвестного при обратном ходе.

Одной из модификаций метода Гаусса является *схема с выбором главного элемента*. Она состоит в том, что требование неравенства нулю диагональных элементов a_{ii}, на которые происходит деление в процессе исключений, заменяется более жестким: из всех оставшихся в i-м столбце элементов нужно выбрать наибольший по модулю и переставить уравнения так, чтобы этот элемент оказался на месте элемента a_{ii}.

Алгоритм выбора главного элемента приведен на рис. 4.3. Он дополняет алгоритм метода Гаусса (см. рис. 4.2) и используется при этом вместо условной конструкции, выполняющей перестановку уравнений в случае равенства нулю элемента a_{ii}.

Здесь введены новые индексы: l — номер наибольшего по абсолютной величине элемента матрицы в столбце с номером i (т. е. среди элементов $a_{ii}, \ldots, a_{mi}, \ldots, a_{ni}$); m — текущий номер элемента, с которым происходит сравнение. Заметим, что диагональные элементы матрицы называются *ведущими* элементами; ведущий элемент a_{ii} — это коэффициент при i-м неизвестном в i-м уравнении на i-м шаге исключения.

В описанной схеме выбор главного элемента осуществляется *по столбцу*. Существуют также схемы с выбором главного элемента *по строке* и *по всей матрице*.

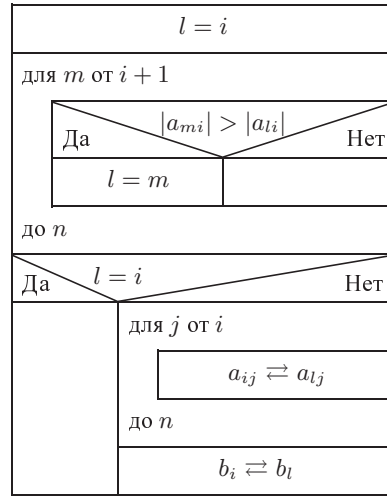


Рис. 4.3. Выбор главного элемента

Благодаря выбору наибольшего по модулю ведущего элемента уменьшаются множители, используемые для преобразования уравнений, что способствует снижению погрешностей вычислений. Поэтому метод Гаусса с выбором главного элемента обеспечивает приемлемую точность решения для не слишком большого числа $(n \lesssim 1000)$ уравнений.

И только для плохо обусловленных систем решения, полученные по этому методу, ненадежны.

Метод Гаусса целесообразно использовать для решения систем с плотно заполненной матрицей. Все элементы матрицы и правые части системы уравнений находятся в оперативной памяти машины. Объем вычислений определяется порядком системы n: число арифметических операций примерно равно $(2/3)n^3$.

Пример. Рассмотрим алгоритм решения линейной системы методом Гаусса и некоторые особенности этого метода для случая трех уравнений:

$$\begin{aligned} 10x_1 - 7x_2 &= 7, \\ -3x_1 + 3x_2 + 6x_3 &= 4, \\ 5x_1 - x_2 + 5x_3 &= 6. \end{aligned}$$

Исключим x_1 из второго и третьего уравнений. Для этого сначала умножим первое уравнение на 0.3 и результат прибавим ко второму, а затем умножим первое же уравнение на -0.5 и результат прибавим к третьему.

Получим

$$\begin{aligned} 10x_1 - 7x_2 &= 7, \\ -0.1x_2 + 6x_3 &= 6.1, \\ 2.5x_2 + 5x_3 &= 2.5. \end{aligned}$$

Прежде чем исключать x_2 из третьего уравнения, заметим, что коэффициент при x_2 во втором уравнении (ведущий элемент) мал; поэтому было бы лучше переставить второе и третье уравнения. Однако мы проводим сейчас вычисления в рамках точной арифметики и погрешности округлений не опасны, поэтому продолжим исключение. Умножим второе уравнение на 25 и результат сложим с третьим уравнением. Получим систему в треугольном виде:

$$\begin{aligned} 10x_1 - 7x_2 &= 7, \\ -0.1x_2 + 6x_3 &= 6.1, \\ 155x_3 &= 155. \end{aligned}$$

На этом заканчивается прямой ход метода Гаусса.

Обратный ход состоит в последовательном вычислении x_3, x_2, x_1 соответственно из третьего, второго, первого уравнений. Проведем эти вычисления:

$$x_3 = \frac{155}{155} = 1, \quad x_2 = \frac{6x_3 - 6.1}{0.1} = -1, \quad x_1 = \frac{7x_2 + 7}{10} = 0.$$

Подстановкой в исходную систему легко убедиться, что $(0, -1, 1)$ и есть ее решение.

Изменим теперь слегка коэффициенты системы таким образом, чтобы сохранить прежним решение и вместе с тем при вычислениях использовать округления. Таким условиям, в частности, соответствует система

$$\begin{aligned} 10x_1 - 7x_2 &= 7, \\ -3x_1 + 2.099x_2 + 6x_3 &= 3.901, \\ 5x_1 - x_2 + 5x_3 &= 6. \end{aligned}$$

Здесь изменены коэффициент при x_2 и правая часть второго уравнения. Будем снова вести процесс исключения, причем вычисления проведем в рамках арифметики с плавающей точкой, сохраняя пять разрядов числа. После первого шага исключения получим

$$\begin{aligned} 10x_1 - 7x_2 &= 7, \\ -0.001x_2 + 6x_3 &= 6.001, \\ 2.5x_2 + 5x_3 &= 2.5. \end{aligned}$$

Следующий шаг исключения проводим при малом ведущем элементе (-0.001). Чтобы исключить x_2 из третьего уравнения, мы вынуждены умножить второе уравнение на 2500. При умножении 6.001 на 2500 получаем число 15 002.5, которое при округлении до пяти разрядов дает 15 003.

При прибавлении к этому числу 2.5 получается число 15 005.5, которое округляется до 15 006. В результате получаем третье уравнение в виде

$$15\,005x_3 = 15\,006.$$

Отсюда $x_3 = 15006/15005 = 1.0001$. Из второго и первого уравнений найдем

$$x_2 = \frac{6.001 - 6 \cdot 1.0001}{-0.001} = -0.4, \quad x_1 = \frac{7 + 7 \cdot (-0.4)}{10} = 0.42.$$

Вычисления проводились с округлением до пяти разрядов по аналогии с процессом вычислений на компьютере. В результате этого было получено решение $(0.42, -0.4, 1.0001)$ вместо $(0, -1, 1)$.

Такая большая неточность результатов объясняется малой величиной ведущего элемента. В подтверждение этому до исключения x_2 из третьего уравнения переставим уравнения системы:

$$\begin{aligned} 10x_1 - 7x_2 &= 7, \\ 2.5x_2 + 5x_3 &= 2.5, \\ -0.001x_2 + 6x_3 &= 6.001. \end{aligned}$$

Исключим теперь x_2 из третьего (бывшего второго) уравнения, прибавив к нему второе, умноженное на 0.0004 (ведущий элемент здесь равен 2.5). Третье уравнение примет вид

$$6.002x_3 = 6.002.$$

Отсюда находим $x_3 = 1$. С помощью второго и первого уравнений вычислим x_2, x_1:

$$x_2 = \frac{2.5 - 5 \cdot 1}{2.5} = -1, \quad x_1 = \frac{7 + 7 \cdot (-1)}{10} = 0.$$

Таким образом, в результате перестановки уравнений, т. е. выбора наибольшего по модулю из оставшихся в данном столбце элементов, погрешность решения в рамках данной точности исчезла.

Рассмотрим подробнее вопрос о погрешностях решения систем линейных уравнений методом Гаусса. Запишем систему в матричном виде: $A\mathbf{x} = \mathbf{b}$. Решение этой системы можно представить в виде $\mathbf{x} = A^{-1}\mathbf{b}$. Однако вычисленное по методу Гаусса решение $\mathbf{x}_*$ отличается от этого решения из-за погрешностей округлений, связанных с ограниченностью разрядной сетки машины.

Существуют две величины, характеризующие степень отклонения полученного решения от точного. Одна из них — *погрешность* $\Delta\mathbf{x}$, равная разности этих значений; другая — *невязка* $\mathbf{r}$, равная разности между левой и правой частями уравнений при подстановке в них решения:

$$\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_*, \quad \mathbf{r} = A\mathbf{x}_* - \mathbf{b}.$$

Можно показать, что если одна из этих величин равна нулю, то и другая должна равняться нулю. Однако из малости одной не следует малость другой. При $\Delta\mathbf{x} \approx 0$ обычно $\mathbf{r} \approx 0$, но обратное утверждение справедливо не всегда. В частности, для плохо обусловленных систем при $\mathbf{r} \approx 0$ погрешность решения может быть большой.

Вместе с тем в практических расчетах, если система не является плохо обусловленной, контроль точности решения осуществляется с помощью невязки (погрешность же обычно вычислить невозможно, поскольку неизвестно точное решение). Можно отметить, что метод Гаусса с выбором главного элемента в этих случаях дает малые невязки.

Понятия погрешности и невязки используются при численном решении не только систем линейных уравнений, но и других задач. В зависимости от задачи погрешность и невязка могут быть величинами скалярными, векторными (как в данном случае), матричными и др.

3. Определитель и обратная матрица. Ранее уже отмечалось, что непосредственное нахождение определителя требует большого объема вычислений. Вместе с тем легко вычисляется определитель треугольной матрицы: он равен произведению ее диагональных элементов.

Для приведения матрицы к треугольному виду может быть использован *метод исключения*, т. е. прямой ход метода Гаусса. В процессе исключения элементов величина определителя не меняется. Знак определителя меняется на противоположный при перестановке его столбцов или строк. Следовательно, значение определителя после приведения матрицы A к треугольному виду вычисляется по формуле

$$\det A = (-1)^k \prod_{i=1}^{n} a_{ii}.$$

Здесь диагональные элементы a_{ii} берутся из преобразованной (а не исходной) матрицы. Через k обозначено число перестановок строк (или столбцов) матрицы при ее приведении к треугольному виду (для получения ненулевого или максимального по модулю ведущего элемента на каждом этапе исключения). Благодаря методу исключения можно вычислять определители 1000-го и большего порядков, и объем вычислений значительно меньший, чем в проведенных ранее оценках.

Теперь найдем обратную матрицу A^{-1}. Обозначим ее элементы через z_{ij}. Запишем равенство $AA^{-1} = E$ в виде

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{n1} & z_{n2} & \dots & z_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}.$$

Отсюда следует, что

$$A\mathbf{z}_j = \mathbf{e}_j, \quad j = 1, 2, \dots, n, \tag{4.12}$$

где $\mathbf{z}_j$ и $\mathbf{e}_j$ — j-е столбцы матриц A^{-1} и E соответственно [1]. Таким образом, для нахождения j-го столбца обратной матрицы нужно решить систему уравнений (4.12). Решив n таких систем для $j = 1, 2, \ldots, n$, мы найдем все столбцы $\mathbf{z}_j$ и, следовательно, саму обратную матрицу.

Поскольку при разных j матрица A системы (4.12) не меняется, исключение неизвестных при использовании метода Гаусса (прямой ход) проводится только один раз, причем сразу для всех правых частей — столбцов $\mathbf{e}_j$. Затем для каждой из систем (4.12) делается обратный ход с соответствующей преобразованной правой частью.

Оценки показывают, что это весьма экономичный способ обращения матрицы. Он требует примерно лишь в три раза больше действий, чем при решении одной системы уравнений.

Степень отклонения вычиленной обратной матрицы A_*^{-1} от ее точного значения характеризуется погрешностью ΔA^{-1} и невязкой R:

$$\Delta A^{-1} = A^{-1} - A_*^{-1}, \quad R = AA_*^{-1} - E.$$

4. Метод прогонки. Он является модификацией метода Гаусса для частного случая разреженных систем — системы уравнений с *трехдиагональной матрицей*. Такие системы получаются при моделировании некоторых инженерных задач, а также при численном решении краевых задач для дифференциальных уравнений.

Запишем систему уравнений в виде

$$\begin{aligned}
&b_1x_1 + c_1x_2 && = d_1,\\
&a_2x_1 + b_2x_2 + c_2x_3 && = d_2,\\
&\qquad a_3x_2 + b_3x_3 + c_3x_4 && = d_3,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\qquad\qquad a_{n-1}x_{n-2} + b_{n-1}x_{n-1} + c_{n-1}x_n && = d_{n-1},\\
&\qquad\qquad\qquad a_nx_{n-1} + b_nx_n && = d_n.
\end{aligned} \tag{4.13}$$

На главной диагонали матрицы этой системы стоят элементы b_1, $b_2, \ldots, b_n$, над ней — элементы $c_1, c_2, \ldots, c_{n-1}$, под ней — элементы a_2, $a_3, \ldots, a_n$ (при этом обычно все коэффициенты b_i не равны нулю). Остальные элементы матрицы равны нулю.

Метод прогонки состоит из двух этапов — *прямой прогонки* (аналога прямого хода метода Гаусса) и *обратной прогонки* (аналога обратного хода метода Гаусса). Прямая прогонка состоит в вычислении прогоночных коэффициентов A_i, B_i, с помощью которых каждое неизвестное x_i выражается через x_{i+1}:

$$x_i = A_ix_{i+1} + B_i, \quad i = 1, 2, \ldots, n-1. \tag{4.14}$$

[1] Заметим, что в столбце $\mathbf{e}_j$ равны нулю все компоненты, кроме j-й, которая равна единице.

Из первого уравнения системы (4.13) найдем

$$x_1 = -\frac{c_1}{b_1}\, x_2 + \frac{d_1}{b_1}\,.$$

С другой стороны, по формуле (4.14) $x_1 = A_1 x_2 + B_1$. Приравнивая коэффициенты в обоих выражениях для x_1, получаем

$$A_1 = -\frac{c_1}{b_1}\,, \quad B_1 = \frac{d_1}{b_1}\,. \tag{4.15}$$

Подставим во второе уравнение системы (4.13) вместо x_1 его выражение через x_2 по формуле (4.14):

$$a_2(A_1 x_2 + B_1) + b_2 x_2 + c_2 x_3 = d_2.$$

Выразим отсюда x_2 через x_3:

$$x_2 = \frac{-c_2 x_3 + d_2 - a_2 B_1}{a_2 A_1 + b_2}\,,$$

или

$$x_2 = A_2 x_3 + B_2,$$
$$A_2 = -\frac{c_2}{e_2}\,, \quad B_2 = \frac{d_2 - a_2 B_1}{e_2}\,, \quad e_2 = a_2 A_1 + b_2.$$

Аналогично вычисляются прогоночные коэффициенты для любого номера i:

$$A_i = -\frac{c_i}{e_i}\,, \quad B_i = \frac{d_i - a_i B_{i-1}}{e_i}\,,$$
$$e_i = a_i A_{i-1} + b_i, \quad i = 2, 3, \ldots, n-1. \tag{4.16}$$

Обратная прогонка состоит в последовательном вычислении неизвестных x_i. Сначала нужно найти x_n. Для этого воспользуемся выражением (4.14) при $i = n-1$ и последним уравнением системы (4.13). Запишем их:

$$x_{n-1} = A_{n-1} x_n + B_{n-1},$$
$$a_n x_{n-1} + b_n x_n = d_n.$$

Отсюда, исключая x_{n-1}, находим

$$x_n = \frac{d_n - a_n B_{n-1}}{b_n + a_n A_{n-1}}\,.$$

Далее, используя формулы (4.14) и вычисленные ранее по формулам (4.15), (4.16) прогоночные коэффициенты, последовательно вычисляем все неизвестные $x_{n-1}, x_{n-2}, \ldots, x_1$. Алгоритм решения системы линейных уравнений вида (4.13) методом прогонки приведен на рис. 4.4.

При анализе алгоритма метода прогонки надо учитывать возможность деления на нуль в формулах (4.15), (4.16). Можно показать, что при выполнении условия преобладания диагональных элементов, т. е. если $|b_i| \geqslant |a_i| + |c_i|$, причем хотя бы для одного значения i имеет место строгое неравенство, деления на нуль не возникает, и система (4.13) имеет единственное решение.

Приведенное условие преобладания диагональных элементов обеспечивает также устойчивость метода прогонки относительно погрешностей округлений. Последнее обстоятельство позволяет использовать метод прогонки для решения больших систем уравнений. Заметим, что данное условие устойчивости прогонки является достаточным, но не необходимым. В ряде случаев для хорошо обусловленных систем вида (4.13) метод прогонки оказывается устойчивым даже при нарушении условия преобладания диагональных элементов.

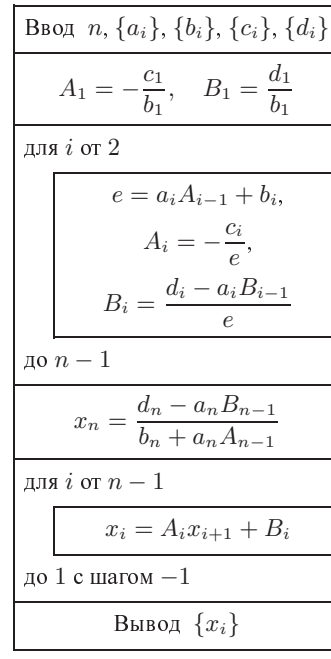


Рис. 4.4. Метод прогонки

5. О других прямых методах. Среди прямых методов наиболее распространен метод Гаусса; он удобен для вычислений на компьютере. Перечислим некоторые другие методы.

Схема Жордана при выборе главного элемента не учитывает коэффициенты тех уравнений, из которых уже выбирался главный элемент. Она не имеет преимуществ по сравнению с методом Гаусса. Отметим лишь, что здесь облегчается обратный ход, поскольку система приводится к диагональному виду (а не к треугольному). Эта схема часто используется для нахождения обратной матрицы.

Метод квадратного корня используется в тех случаях, когда матрица системы является симметричной.

Метод оптимального исключения удобен при построчном вводе матрицы системы в оперативную память. Однако построчный ввод имеет и недостатки: частые обращения к внешним устройствам, невозможность выбора главного элемента и др.

Клеточные методы могут использоваться для решения больших систем, когда матрица и вектор правых частей целиком не помещаются в оперативной памяти.

Эти и другие методы решения систем линейных уравнений подробно описаны в более полных пособиях по численным методам, а также в специальной литературе по линейной алгебре (см. список литературы).

§ 3. Итерационные методы

1. Уточнение решения. Решения, получаемые с помощью прямых методов, обычно содержат погрешности, вызванные округлениями при выполнении операций над числами с плавающей точкой на компьютере с ограниченным числом разрядов. В ряде случаев эти погрешности могут быть значительными, и необходимо найти способ их уменьшения. Рассмотрим здесь один из методов, позволяющий уточнить решение, полученное с помощью прямого метода.

Найдем решение системы линейных уравнений

$$A\mathbf{x} = \mathbf{b}. \tag{4.17}$$

Пусть с помощью некоторого прямого метода вычислено приближенное решение $\mathbf{x}^{(0)}$ (т. е. приближенные значения неизвестных $x_1^{(0)}, x_2^{(0)}, \ldots$ $\ldots, x_n^{(0)}$), называемое *начальным* или *нулевым приближением* к решению. Подставляя это решение в левую часть системы (4.17), получаем некоторый столбец правых частей $\mathbf{b}^{(0)}$, отличный от $\mathbf{b}$:

$$A\mathbf{x}^{(0)} = \mathbf{b}^{(0)}. \tag{4.18}$$

Введем обозначения: $\Delta\mathbf{x}^{(0)}$ — погрешность полученного решения, $\mathbf{r}^{(0)}$ — невязка, т. е.

$$\Delta\mathbf{x}^{(0)} = \mathbf{x} - \mathbf{x}^{(0)}, \quad \mathbf{r}^{(0)} = A\mathbf{x}^{(0)} - \mathbf{b} = \mathbf{b}^{(0)} - \mathbf{b}. \tag{4.19}$$

Вычитая равенство (4.18) из равенства (4.17), с учетом обозначений (4.19) получаем

$$A\Delta\mathbf{x}^{(0)} = -\mathbf{r}^{(0)}. \tag{4.20}$$

Решая эту систему, находим значение погрешности $\Delta\mathbf{x}^{(0)}$, которое используем в качестве поправки к приближенному решению $\mathbf{x}^{(0)}$, вычисляя таким образом новое приближенное решение $\mathbf{x}^{(1)}$ (или *следующее приближение* к решению):

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \Delta\mathbf{x}^{(0)}.$$

Таким же способом можно найти новую поправку к решению $\Delta\mathbf{x}^{(1)}$ и следующее приближение $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \Delta\mathbf{x}^{(1)}$ и т. д. Процесс продолжается до тех пор, пока очередное значение погрешности (поправки) $\Delta\mathbf{x}^{(k)}$ не станет достаточно малым, т. е. пока очередные приближенные значения неизвестных $x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_n^{(k+1)}$ не будут мало отличаться от предыдущих значений $x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}$.

Рассмотренный процесс уточнения решения представляет собой фактически итерационный метод решения системы линейных уравнений. При этом заметим, что для нахождения очередного приближения, т. е. на каждой итерации, решаются системы уравнений вида (4.20) с одной и той же матрицей, являющейся матрицей исходной системы (4.17), при

разных правых частях. Это позволяет строить экономичные алгоритмы. Например, при использовании метода Гаусса сокращается объем вычислений на этапе прямого хода.

Решение систем линейных уравнений с помощью рассмотренного метода (а также решение систем линейных уравнений иными итерационными методами, решение итерационными методами уравнений другого вида и их систем) сводится к следующему (рис. 4.5). Вводятся исходные данные, например, коэффициенты уравнений и допустимое значение погрешности. Необходимо также задать начальные приближения значений неизвестных (вектор-столбец $\mathbf{x}^{(0)}$). Они либо вводятся в компьютер, либо вычисляются каким-либо способом (в частности, путем решения системы уравнений с помощью прямого метода). Затем организуется циклический вычислительный процесс, каждый цикл которого представляет собой одну итерацию — переход от предыдущего приближения $\mathbf{x}^{(k-1)}$ к последующему $\mathbf{x}^{(k)}$. Если оказывается, что с увеличением числа итераций приближенное решение стремится к точному:

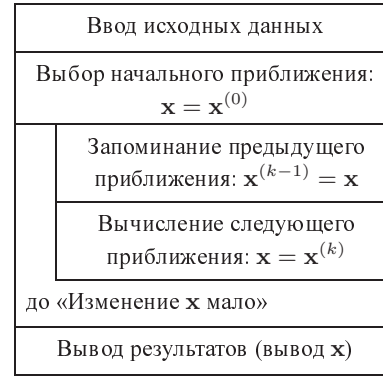


Рис. 4.5. Решение системы уравнений методом итераций

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x},$$

то итерационный метод называют *сходящимся*.

На практике наличие сходимости и достижение требуемой точности обычно определяют приближенно, поступая следующим образом. При малом (с заданной допустимой погрешностью) изменении $\mathbf{x}$ на двух последовательных итерациях, т. е. при малом отличии $\mathbf{x}^{(k)}$ от $\mathbf{x}^{(k-1)}$, процесс прекращается, и происходит вывод значений неизвестных, полученных на последней итерации.

Возможны разные подходы к определению малости отличия $\mathbf{x}$ на двух последовательных итерациях. Например, если задана допустимая погрешность $\varepsilon > 0$, то критерием окончания итерационного процесса можно считать выполнение одного из трех неравенств:

$$\left|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\right| = \sqrt{\sum_{i=1}^{n} \left(x_i^{(k)} - x_i^{(k-1)}\right)^2} < \varepsilon, \tag{4.21}$$

$$\max_{1\leqslant i\leqslant n} \left|x_i^{(k)} - x_i^{(k-1)}\right| < \varepsilon, \tag{4.22}$$

$$\max_{1\leqslant i\leqslant n}\left|\frac{x_i^{(k)}-x_i^{(k-1)}}{x_i^{(k)}}\right|<\varepsilon,\quad \text{при}\quad |x_i|\gg 1. \tag{4.23}$$

Здесь в первом случае отличие векторов $\mathbf{x}^{(k)}$ и $\mathbf{x}^{(k-1)}$ «на ε» понимается в смысле малости модуля их разности, во втором — в смысле малости разностей всех соответствующих компонент векторов, в третьем — в смысле малости относительных разностей компонент. Если система не является плохо обусловленной, то в качестве критерия окончания итерационного процесса можно использовать и условие малости невязки, например

$$\left|\mathbf{r}^{(k)}\right|<\varepsilon. \tag{4.24}$$

Заметим, что в рассмотренном алгоритме не предусмотрен случай отсутствия сходимости. Для предотвращения непроизводительных затрат машинного времени в алгоритм вводят счетчик числа итераций и при достижении им некоторого заданного значения счет прекращают. Такой элемент будет в дальнейшем введен в структурограмму.

2. Метод простой итерации. Этот метод широко используется для численного решения уравнений и их систем различных видов. Рассмотрим применение метода простой итерации к решению систем линейных уравнений.

Запишем исходную систему уравнений в векторно-матричном виде (4.2) и выполним ряд тождественных преобразований:

$$\begin{gathered}A\mathbf{x}=\mathbf{b};\quad \mathbf{0}=\mathbf{b}-A\mathbf{x};\quad \mathbf{x}=\mathbf{b}-A\mathbf{x}+\mathbf{x};\\ \mathbf{x}=(\mathbf{b}-A\mathbf{x})\tau+\mathbf{x};\quad \mathbf{x}=(E-\tau A)\mathbf{x}+\tau\mathbf{b};\\ \mathbf{x}=B\mathbf{x}+\tau\mathbf{b},\end{gathered} \tag{4.25}$$

где $\tau\neq 0$ — некоторое число, E — единичная матрица, $B=E-\tau A$. Получившаяся система (4.25) эквивалентна исходной системе и служит основой для построения метода простой итерации.

Выберем некоторое начальное приближение $\mathbf{x}^{(0)}$ и подставим его в правую часть системы (4.25):

$$\mathbf{x}^{(1)}=B\mathbf{x}^{(0)}+\tau\mathbf{b}.$$

Поскольку $\mathbf{x}^{(0)}$ не является решением системы, в левой части (4.25) получится некоторый столбец $\mathbf{x}^{(1)}$, в общем случае отличный от $\mathbf{x}^{(0)}$. Полученный столбец $\mathbf{x}^{(1)}$ будем рассматривать в качестве следующего (первого) приближения к решению. Аналогично, по известному k – му приближению можно найти $(k+1)$ – е приближение:

$$\mathbf{x}^{(k+1)}=B\mathbf{x}^{(k)}+\tau\mathbf{b},\quad k=0,1,2,\ldots \tag{4.26}$$

Формула (4.26) и выражает собой метод простой итерации. Для ее применения нужно задать неопределенный пока параметр τ. От значения τ зависит, будет ли сходиться метод, а если будет, то какова будет *скорость*

сходимости, т. е. как много итераций нужно совершить для достижения требуемой точности. В частности, справедлива следующая теорема.

Т е о р е м а. *Пусть* $\det A \neq 0$. *Метод простой итерации* (4.26) *сходится тогда и только тогда, когда все собственные числа* [1] *матрицы* $B = A - \tau E$ *по модулю меньше единицы.*

Для некоторых типов матрицы A можно указать правило выбора τ, обеспечивающее сходимость метода и оптимальную скорость сходимости. В простейшем же случае τ можно положить равным некоторому постоянному числу, например, 1, 0.1 и т. д.

3. Метод Гаусса–Зейделя. Одним из самых распространенных итерационных методов, отличающийся простотой и легкостью программирования, является *метод Гаусса–Зейделя*.

Проиллюстрируем сначала этот метод па примере решения системы

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \tag{4.27}$$

Предположим, что диагональные элементы a_{11}, a_{22}, a_{33} отличны от нуля (в противном случае можно переставить уравнения). Выразим неизвестные x_1, x_2 и x_3 соответственно из первого, второго и третьего уравнений системы (4.27):

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3), \tag{4.28}$$

$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3), \tag{4.29}$$

$$x_3 = \frac{1}{a_{33}}(b_3 - a_{31}x_1 - a_{32}x_2). \tag{4.30}$$

Зададим некоторые начальные (нулевые) приближения значений неизвестных: $x_1 = x_1^{(0)}$, $x_2 = x_2^{(0)}$, $x_3 = x_3^{(0)}$. Подставляя эти значения в правую часть выражения (4.28), получаем новое (первое) приближение для x_1:

$$x_1^{(1)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}\right).$$

Используя это значение для x_1 и приближение $x_3^{(0)}$ для x_3, находим из (4.29) первое приближение для x_2:

$$x_2^{(1)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)}\right).$$

И наконец, используя вычисленные значения $x_1 = x_1^{(1)}$, $x_2 = x_2^{(1)}$, находим

[1] См. § 4.

с помощью выражения (4.30) первое приближение для x_3:

$$x_3^{(1)} = \frac{1}{a_{33}}\big(b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}\big).$$

На этом заканчивается первая итерация решения системы (4.28)–(4.30). Используя теперь значения $x_1^{(1)}$, $x_2^{(1)}$, $x_3^{(1)}$, можно таким же способом провести вторую итерацию, в результате которой будут найдены вторые приближения к решению: $x_1 = x_1^{(2)}$, $x_2 = x_2^{(2)}$, $x_3 = x_3^{(2)}$ и т. д.

Приближение с номером k можно вычислить, зная приближение с номером $k-1$, как

$$\begin{aligned}
x_1^{(k)} &= \frac{1}{a_{11}}\big(b_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)}\big),\\
x_2^{(k)} &= \frac{1}{a_{22}}\big(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)}\big),\\
x_3^{(k)} &= \frac{1}{a_{33}}\big(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}\big).
\end{aligned}$$

Итерационный процесс продолжается до тех пор, пока значения $x_1^{(k)}$, $x_2^{(k)}$, $x_3^{(k)}$ не станут близкими с заданной погрешностью к значениям $x_1^{(k-1)}$, $x_2^{(k-1)}$, $x_3^{(k-1)}$.

Пример. Решить с помощью метода Гаусса–Зейделя следующую систему уравнений:

$$\begin{aligned}
4x_1 - \ x_2 + \ x_3 &= 4,\\
2x_1 + 6x_2 - \ x_3 &= 7,\\
x_1 + 2x_2 - 3x_3 &= 0.
\end{aligned}$$

Легко проверить, что решение данной системы следующее: $x_1 = x_2 = x_3 = 1$.

Решение. Выразим неизвестные x_1, x_2 и x_3 соответственно из первого, второго и третьего уравнений:

$$x_1 = \frac{1}{4}\,(4 + x_2 - x_3), \quad x_2 = \frac{1}{6}\,(7 - 2x_1 + x_3),$$
$$x_3 = \frac{1}{3}\,(x_1 + 2x_2).$$

В качестве начального приближения (как это обычно делается) примем $x_1^{(0)} = 0$, $x_2^{(0)} = 0$, $x_3^{(0)} = 0$. Найдем новые приближения неизвестных:

$$x_1^{(1)} = \frac{1}{4}(4 + 0 - 0) = 1, \quad x_2^{(1)} = \frac{1}{6}(7 - 2 \cdot 1 + 0) = \frac{5}{6}\,,$$
$$x_3^{(1)} = \frac{1}{3}\left(1 + 2 \cdot \frac{5}{6}\right) = \frac{8}{9}\,.$$

Аналогично вычислим следующие приближения:

$$x_1^{(2)} = \frac{1}{4}\left(4 + \frac{5}{6} - \frac{8}{9}\right) = \frac{71}{72}, \quad x_2^{(2)} = \frac{1}{6}\left(7 - 2\cdot\frac{71}{72} + \frac{8}{9}\right) = \frac{71}{72},$$
$$x_3^{(2)} = \frac{1}{3}\left(\frac{71}{72} + 2\cdot\frac{71}{72}\right) = \frac{71}{72}.$$

Итерационный процесс можно продолжать до получения малой разности между значениями неизвестных в двух последовательных итерациях.

Рассмотрим теперь систему n линейных уравнений с n неизвестными. Запишем ее в виде

$$a_{i1}x_1 + \ldots + a_{i,i-1}x_{i-1} + a_{ii}x_i + a_{i,i+1}x_{i+1} + \ldots + a_{in}x_n = b_i,$$
$$i = 1, 2, \ldots, n.$$

Здесь также будем, предполагать, что все диагональные элементы отличны от нуля. Тогда в соответствии с методом Гаусса–Зейделя k-е приближение к решению можно представить в виде

$$x_i^{(k)} = \frac{1}{a_{ii}}\Big(b_i - a_{i1}x_1^{(k)} - \ldots - a_{i,i-1}x_{i-1}^{(k)} - \\ - a_{i,i+1}x_{i+1}^{(k-1)} - \ldots - a_{in}x_n^{(k-1)}\Big), \quad i = 1, 2, \ldots, n. \quad (4.31)$$

Итерационный процесс продолжается до тех пор, пока все значения $x_i^{(k)}$ не станут близкими к $x_i^{(k-1)}$, т. е. в качестве критерия завершения итераций используется одно из условий (4.21) – (4.23), (4.24).

Для сходимости итерационного процесса (4.31) достаточно, чтобы модули диагональных коэффициентов для каждого уравнения системы были не меньше сумм модулей всех остальных коэффициентов (преобладание диагональных элементов):

$$|a_{ii}| \geqslant \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, \ldots, n. \quad (4.32)$$

При этом хотя бы для одного уравнения неравенство должно выполняться строго. Эти условия являются достаточными для сходимости метода, но они не являются необходимыми, т. е. для некоторых систем итерации сходятся и при нарушении условий (4.32).

Алгоритм решения системы n линейных уравнений методом Гаусса–Зейделя представлен на рис. 4.6. В качестве исходных данных вводятся n, коэффициенты и правые части уравнений системы, погрешность ε, максимально допустимое число итераций M, а также начальные приближения переменных x_i ($i = 1, 2, \ldots, n$). Отметим, что начальные приближения можно не вводить в компьютер, а полагать их равными некоторым значениям (например, нулю). Критерием завершения итераций выбрано

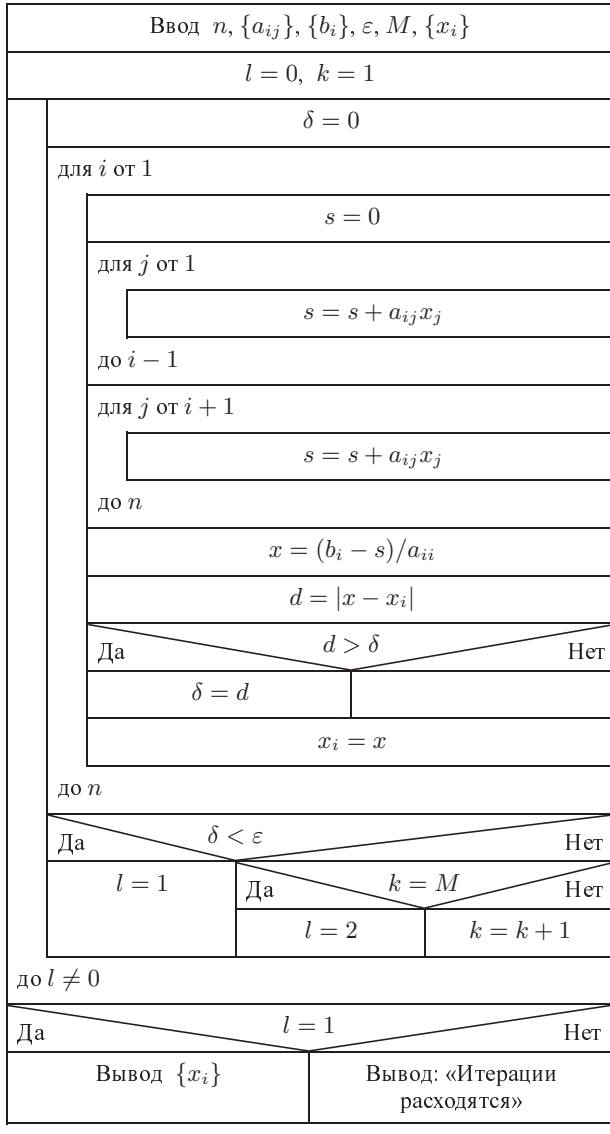


Рис. 4.6. Метод Гаусса–Зейделя

условие (4.22), в котором через δ обозначена максимальная абсолютная величина разности $x_i^{(k)}$ и $x_i^{(k-1)}$:

$$\delta = \max_{1 \leqslant i \leqslant n} \left| x_i^{(k)} - x_i^{(k-1)} \right| < \varepsilon.$$

Для удобства чтения структурограммы объясним другие обозначения: k — порядковый номер итерации; i — номер уравнения, а также переменного, которое вычисляется в соответствующем цикле; j — номер члена вида $a_{ij}x_j^{(k)}$ или $a_{ij}x_j^{(k-1)}$ в правой части соотношения (4.31). Итерационный процесс прекращается либо при $\delta < \varepsilon$, либо при $k = M$. В последнем случае итерации не сходятся, о чем выдается сообщение. Для завершения цикла, реализующего итерационный процесс, используется переменная l, которая принимает значения 0, 1 и 2 соответственно при продолжении итераций, при выполнении условия $\delta < \varepsilon$ и при выполнении условия $k = M$.

§ 4. Задачи на собственные значения

1. Основные понятия. Большое число научно-технических задач, а также некоторые исследования в области вычислительной математики требуют нахождения собственных значений и собственных векторов матриц. Введем некоторые определения, необходимые для изложения материала данного параграфа.

Рассмотрим, квадратную матрицу n-го порядка

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}. \tag{4.33}$$

Вектор $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ называется *собственным вектором* матрицы A, соответствующим *собственному значению* λ, если он удовлетворяет системе уравнений

$$A\mathbf{x} = \lambda\mathbf{x}. \tag{4.34}$$

Поскольку при умножении собственного вектора на скаляр он остается собственным вектором той же матрицы, его можно нормировать. В частности, каждую координату собственного вектора можно разделить на максимальную из них или на длину вектора; в последнем случае получится единичный собственный вектор.

Характеристической матрицей C данной матрицы A называется матрица вида

$$C = A - \lambda E = \begin{pmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{pmatrix}, \tag{4.35}$$

где E — единичная матрица. Легко видеть, что систему (4.34) можно записать в виде

$$(A - \lambda E)\mathbf{x} = \mathbf{0} \quad \text{или} \quad C\mathbf{x} = \mathbf{0}. \tag{4.36}$$

Если перейти к координатной форме записи вектора $\mathbf{x}$, то с учетом (4.33) систему (4.36) можно записать в виде

$$\begin{aligned} (a_{11}-\lambda)x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= 0,\\ a_{21}x_1 + (a_{22}-\lambda)x_2 + \ldots + a_{2n}x_n &= 0,\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots &\\ a_{n1}x_1 + a_{n2}x_2 + \ldots + (a_{nn}-\lambda)x_n &= 0. \end{aligned} \tag{4.37}$$

Система (4.36) или (4.37) является однородной системой n линейных уравнений с n неизвестными. Она имеет ненулевые решения лишь тогда, когда ее определитель равен нулю: $\det C = 0$, причем решение не единственно.

Определитель матрицы C является многочленом n-й степени относительно λ:

$$\det C = c_0\lambda^n + c_1\lambda^{n-1} + \ldots + c_{n-1}\lambda + c_n, \tag{4.38}$$

называемым *характеристическим многочленом*. Корни этого многочлена являются собственными значениями матрицы A.

Для нахождения собственных векторов матрицы требуется решить систему линейных алгебраических уравнений, решение которой не единственно. Из линейной алгебры известно, что в этом случае структура общего решения системы имеет следующий вид: одно или несколько неизвестных, называемых *свободными*, могут принимать любые значения, а остальные неизвестные выражаются через свободные. Число свободных неизвестных равно числу уравнений системы, являющихся следствием остальных уравнений. На практике, если свободное неизвестное одно (что часто и бывает), его полагают равным некоторому числу, например единице. После этого остальные неизвестные (компоненты вектора) находятся однозначно из подсистемы линейно независимых уравнений, в которой отброшено уравнение, являющееся следствием остальных. Эта процедура не влияет на результат решения задачи, поскольку, как уже отмечалось, собственные векторы находятся с точностью до постоянного множителя.

Пример. Вычислить собственные числа и собственные векторы матрицы

$$A = \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}.$$

Решение. Составим характеристический многочлен

$$\begin{vmatrix} 3-\lambda & 1 \\ 2 & 4-\lambda \end{vmatrix} = (3-\lambda)(4-\lambda) - 2 = \lambda^2 - 7\lambda + 10.$$

Найдем корни этого многочлена второй степени:

$$\lambda^2 - 7\lambda + 10 = 0, \quad \lambda_1 = 2, \quad \lambda_2 = 5.$$

Для нахождения собственных векторов $\mathbf{x}_1$, $\mathbf{x}_2$, соответствующих собственным значениям λ_1, λ_2, составим системы уравнений типа (4.36), (4.37) для каждого из них.

При $\lambda_1 = 2$ получим

$$\begin{pmatrix} 3-2 & 1 \\ 2 & 4-2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

или, в координатной форме,

$$\begin{aligned} x_1 + \ x_2 &= 0, \\ 2x_1 + 2x_2 &= 0. \end{aligned}$$

Замечаем, что уравнения линейно зависимы. Поэтому оставляем лишь одно из них.

Из первого уравнения следует, что $x_2 = -x_1$. Неизвестное x_1 можно считать свободным, полагаем $x_1 = 1$. Тогда $x_2 = -1$, и собственный вектор, соответствующий собственному значению $\lambda_1 = 2$, имеет вид $\mathbf{x}_1 = \{1, -1\}$ или $\mathbf{x}_1 = e_1 - e_2$, где e_1, e_2 — единичные орты выбранной базисной системы.

Аналогично находим второй собственный вектор, соответствующий собственному значению $\lambda_2 = 5$. Опуская комментарии, получаем

$$\begin{pmatrix} 3-5 & 1 \\ 2 & 4-5 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \begin{aligned} -2x_1 + x_2 &= 0, \\ 2x_1 - x_2 &= 0. \end{aligned}$$

Отсюда $x_1 = 1$, $x_2 = 2$, $\mathbf{x}_2 = e_1 + 2e_2$.

Вектор $\mathbf{x}_1$ нормирован; нормируем также вектор $\mathbf{x}_2$, разделив его компоненты на наибольшую из них. Получим $\mathbf{x}_2 = 0.5e_1 + e_2$. Можно также привести векторы к единичной длине, разделив их компоненты на значения модулей векторов. В этом случае

$$\mathbf{x}_1 = \frac{1}{\sqrt{2}}(e_1 - e_2), \quad \mathbf{x}_2 = \frac{1}{\sqrt{5}}(e_1 + 2e_2).$$

Мы рассмотрели простейший пример вычисления собственных значений и собственных векторов для матрицы второго порядка. Нетрудно также провести подобное решение задачи для матрицы третьего порядка и для некоторых весьма специальных случаев.

В общем случае, особенно для матриц высокого порядка, задача о нахождении их собственных значений и собственных векторов, называемая *полной проблемой собственных значений*, значительно более сложная.

На первый взгляд может показаться, что вопрос сводится к вычислению корней многочлена (4.38). Однако здесь задача осложнена тем, что среди собственных значений часто встречаются кратные. И кроме того, для произвольной матрицы непросто вычислить сами коэффициенты характеристического многочлена.

Отметим некоторые *свойства собственных значений* для частных типов исходной матрицы.

1. Все собственные значения симметрической матрицы действительны.

2. Если собственные значения матрицы действительны и различны, то соответствующие им собственные векторы ортогональны и образуют базис рассматриваемого пространства. Следовательно, любой вектор в данном пространстве можно выразить через совокупность линейно независимых собственных векторов.

3. Если две матрицы A и B *подобны*, т. е. они связаны соотношением

$$B = P^{-1}AP, \tag{4.39}$$

то их собственные значения совпадают (здесь P — некоторая матрица).

Преобразование подобия (4.39) можно использовать для упрощения исходной матрицы, а задачу о вычислении ее собственных значений свести к аналогичной задаче для более простой матрицы. Очевидно, самым лучшим упрощением матрицы (4.33) было бы приведение ее к треугольному виду

$$A' = \begin{pmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ & a'_{22} & \dots & a'_{2n} \\ & 0 & \ddots & \vdots \\ & & & a'_{nn} \end{pmatrix}.$$

Тогда матрица (4.35) также имела бы треугольный вид. Как известно, определитель треугольной матрицы равен произведению ее диагональных элементов, поэтому характеристический многочлен (4.38) в этом случае имеет вид

$$\det C = (a'_{11} - \lambda)(a'_{22} - \lambda)\dots(a'_{nn} - \lambda).$$

Собственные значения матрицы, равные корням этого многочлена, можно получить сразу:

$$\lambda_1 = a'_{11}, \quad \lambda_2 = a'_{22}, \quad \dots, \quad \lambda_n = a'_{nn}.$$

Таким образом, собственные значения треугольной матрицы равны ее диагональным элементам. То же самое, естественно, относится и к диагональной матрице, которая является частным случаем треугольной.

Некоторые типы матриц удается привести к треугольному виду с помощью преобразования подобия. В частности, симметрическую матрицу можно привести к диагональному виду. На практике часто используется приведение симметрической матрицы к трехдиагональному виду. Процедура вычисления собственных значений для полученной матрицы значительно упрощается по сравнению с задачей для исходной матрицы.

Существует ряд методов, основанных на использовании преобразования подобия, позволяющего привести исходную матрицу к более простой структуре. Мы рассмотрим ниже один из них — метод вращений.

2. Метод вращений. Одним из эффективных методов, позволяющих привести исходную симметрическую матрицу n-го порядка к трехдиагональному виду, является *метод вращений*. Он основан на специально подбираемом вращении системы координат в n-мерном пространстве. Поскольку любое вращение можно заменить последовательностью элементарных (плоских) вращений, то решение задачи можно разбить на ряд шагов, на каждом из которых осуществляется плоское вращение. Таким образом, на каждом шаге выбираются две оси — i-я и j-я ($i \neq j$), и поворот на угол φ производится в плоскости, проходящей через эти оси; остальные оси координат на данном шаге неподвижны. *Матрица вращения* при этом имеет вид

$$P_{ij} = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & 0 & \\ & & \ddots & & & & & & \\ & & & p_{ii} & \dots & p_{ij} & & & \\ & & & \vdots & \ddots & \vdots & & & \\ & & & p_{ji} & \dots & p_{jj} & & & \\ & & & & & & \ddots & & \\ & & & & & & & 1 & \\ & 0 & & & & & & & 1 \end{pmatrix}, \qquad (4.40)$$

$$p_{ii} = p_{jj} = p, \quad p_{ij} = -p_{ji} = -q, \quad p = \cos\varphi, \quad q = \sin\varphi.$$

Здесь мы рассматриваем матрицы с вещественными элементами. В случае комплексных векторов для использования этого метода нужно изменить формулы (4.40).

Для осуществления преобразования подобия (4.39) необходимо найти обратную матрицу P_{ij}^{-1}. Можно показать, что она равна в рассматриваемом случае транспонированной матрице P_{ij}^{T}. Для получения обратной матрицы достаточно провести зеркальное отражение всех элементов исходной матрицы относительно ее диагонали. Другими словами, нужно поменять местами строки и столбцы исходной матрицы; элементы p_{ij} и p_{ji} при этом поменяются местами.

Угол поворота φ на каждом шаге выбирается таким, чтобы в преобразованной матрице обратился в нуль один элемент (в симметрической матрице — два). Процесс преобразования исходной матрицы путем элементарного вращения на любом k-м шаге можно представить в виде рекуррентных соотношений [1)]

$$A^{(k)} = P_{ij}^{\mathrm{T}} A^{(k-1)} P_{ij}, \quad k = 1, 2, \dots \qquad (4.41)$$

[1)] Здесь матрицы вращений P и значения i, j зависят от номера шага k.

Рассмотрим первый шаг преобразования. Сначала вычисляется произведение матриц $B = A^{(0)}P_{ij}$ (здесь $A^{(0)}$ — исходная матрица A). Как видно из (4.40), в полученной матрице отличными от исходных являются элементы, стоящие в i-м и j-м столбцах; остальные элементы совпадают с элементами матрицы $A^{(0)}$, т. е.

$$\begin{gathered} b_{li} = a_{li}^{(0)}p + a_{lj}^{(0)}q, \quad b_{lj} = -a_{li}^{(0)}q + a_{lj}^{(0)}p, \\ b_{lm} = a_{lm}^{(0)}, \quad m \neq i, j, \quad l = 1, 2, \ldots, n. \end{gathered} \tag{4.42}$$

Затем находится преобразованная матрица $A^{(1)} = P_{ij}^{\mathrm{T}}B$. Элементы полученной матрицы отличаются от элементов матрицы B только i-й и j-й строками. Они связаны соотношениями

$$\begin{gathered} a_{im}^{(1)} = b_{im}p + b_{jm}q, \quad a_{jm}^{(1)} = -b_{im}q + b_{jm}p, \\ a_{lm}^{(1)} = b_{lm}, \quad l \neq i, j, \quad m = 1, 2, \ldots, n. \end{gathered} \tag{4.43}$$

Таким образом, преобразованная матрица $A^{(1)}$ отличается от $A^{(0)}$ элементами строк и столбцов с номерами i и j. Эти элементы пересчитываются по формулам (4.42), (4.43). В данных формулах пока не определенными остались параметры p и q; при этом лишь один из них свободный, поскольку они подчиняются тождеству

$$p^2 + q^2 = 1. \tag{4.44}$$

Недостающее одно уравнение для определения этих параметров получается из условия обращения в нуль некоторого элемента новой матрицы $A^{(1)}$. В зависимости от выбора этого элемента строятся различные алгоритмы метода вращений.

Одним из таких алгоритмов является последовательное обращение в нуль всех ненулевых элементов, лежащих вне трех диагоналей исходной симметрической матрицы. Это так называемый *прямой метод вращений*. В соответствии с этим методом обращение в нуль элементов матрицы производится последовательно, начиная с элементов первой строки (и первого столбца, так как матрица симметрическая).

Процесс вычислений поясним с использованием схематического изображения матрицы (рис. 4.7). Точками отмечены элементы матрицы. Наклонные линии указывают три диагонали матрицы, элементы на которых после окончания расчета отличны от нуля. Алгоритм решения задачи нужно построить таким образом, чтобы все элементы по одну сторону от этих трех диагоналей обратились в нуль; тогда симметрично расположенные элементы также станут нулевыми. Обращение элементов в нуль можно выполнять, например, в следующей последовательности: $a_{13}, a_{14}, \ldots, a_{1n}$, $a_{24}, a_{25}, \ldots, a_{2n}, \ldots, a_{n-2,n}$.

Рассмотрим сначала первый шаг данного метода, состоящий в обращении в нуль элемента a_{13} (и автоматически a_{31}). Для осуществления

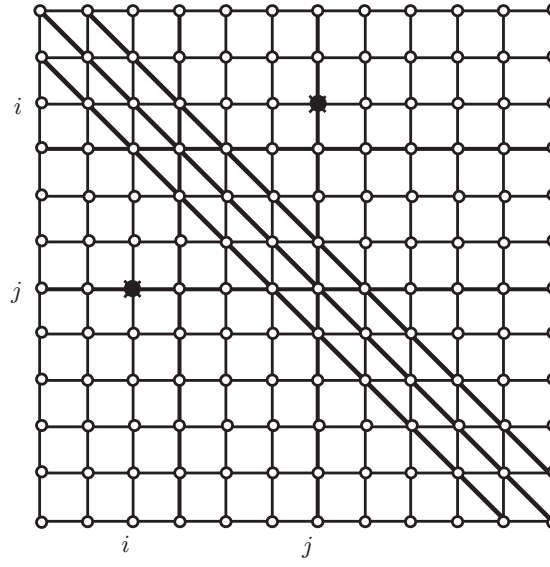


Рис. 4.7

элементарного вращения нужно выбрать две оси — i-ю и j-ю, так чтобы элемент a_{13} оказался в строке или столбце с номером i или j. Положим $i = 2$, $j = 3$ и умножим матрицу $A^{(0)}$ справа на матрицу вращения P_{23} и слева на транспонированную матрицу P_{23}^{T}. Получим новые значения элементов матрицы, которые вычисляются по формулам (4.42), (4.43). Полагая в них $l = 1$ и $m = 3$, находим

$$a_{13}^{(1)} = b_{13} = -a_{12}q + a_{13}p = 0.$$

Учитывая тождество (4.44), получаем систему уравнений для определения параметров p, q:

$$\begin{aligned} a_{13}p - a_{12}q &= 0, \\ p^2 + q^2 &= 1. \end{aligned}$$

Решая эту систему, находим

$$p = \frac{a_{12}}{\sqrt{a_{12}^2 + a_{13}^2}}, \quad q = \frac{a_{13}}{\sqrt{a_{12}^2 + a_{13}^2}}.$$

Используя эти параметры p, q, можно по формулам (4.42), (4.43) вычислить значения элементов, стоящих в строках и столбцах с номерами $i = 2, 3$; $j = 2, 3$ (остальные элементы исходной матрицы не изменились).

Аналогично, выбирая для элементарного вращения i-ю и j-ю оси, можно добиться нулевого значения любого элемента $a_{i-1,j}^{(k)}$ на k-м шаге. В этом

случае строится матрица вращения P_{ij}, параметры которой вычисляются по формулам, полученным из условия равенства нулю элемента $a_{i-1,j}^{(k)}$ и (4.44). Эти формулы имеют вид

$$p = \frac{a_{i-1,i}^{(k-1)}}{\sqrt{\left(a_{i-1,i}^{(k-1)}\right)^2 + \left(a_{i-1,j}^{(k-1)}\right)^2}}, \quad q = \frac{a_{i-1,j}^{(k-1)}}{\sqrt{\left(a_{i-1,i}^{(k-1)}\right)^2 + \left(a_{i-1,j}^{(k-1)}\right)^2}}.$$

Учитывая найденные значения параметров p, q, можно по формулам (4.42), (4.43) найти элементы преобразованной матрицы. Для иллюстрации вновь обратимся к рис. 4.7. Вертикальными линиями показаны столбцы с номерами i и j, соответствующими осям элементарного вращения, горизонтальными — строки с теми же номерами. На рассматриваемом шаге матрица преобразуется таким образом, чтобы отмеченные крестиками элементы обратились в нуль. Элементарное вращение (4.41) на каждом шаге требует пересчета всех элементов отмеченных столбцов и строк. Учитывая симметрию, можно вычислить лишь все элементы столбцов, а элементы получаются из условий симметрии. Исключение составляют лишь элементы, расположенные на пересечениях этих строк и столбцов. Они изменяются на каждом из двух этапов выполняемого шага.

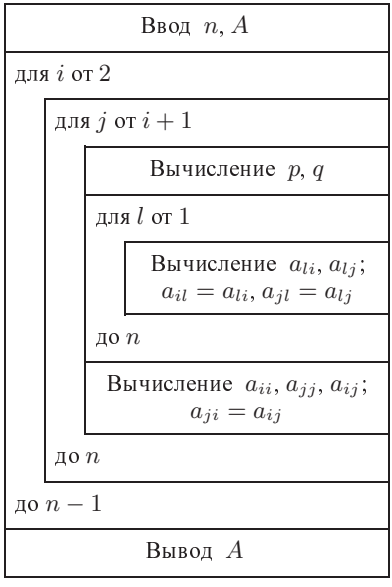


Рис. 4.8. Метод вращений

Таким образом, на каждом шаге преобразования симметрической матрицы для вычисления элементов столбцов используются формулы (4.42), а элементы, находящиеся на пересечениях изменяемых строк и столбцов, пересчитываются еще по формулам (4.43). При этом полученные ранее нулевые элементы не изменяются. Алгоритм приведения симметрической матрицы к трехдиагональному виду с помощью прямого метода вращений представлен на рис. 4.8.

Собственные значения полученной трехдиагональной матрицы будут также собственными значениями исходной матрицы. Собственные векторы $\mathbf{x}_i$ исходной матрицы не равны непосредственно собственным векторам $\mathbf{y}_i$ трехдиагональной матрицы, а вычисляются с помощью соотношений

$$\mathbf{x}_i = P_{23}P_{24}\ldots P_{n-1,n}\,\mathbf{y}_i. \tag{4.45}$$

3. Трехдиагональные матрицы. Как было показано в п. 2, симметрическую матрицу можно привести с помощью преобразований подобия к трехдиагональному виду. Кроме того, трехдиагональные матрицы представляют самостоятельный интерес, поскольку они встречаются в вычислительной практике, и нередко требуется находить их собственные значения и собственные векторы. Рассмотрим трехдиагональную матрицу вида

$$A = \begin{pmatrix} b_1 & c_1 & & & & 0 \\ a_2 & b_2 & c_2 & & & \\ \dots & \dots & \dots & \dots & \dots & \dots \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & & & & a_n & b_n \end{pmatrix}. \tag{4.46}$$

Здесь элементы $b_1, b_2, \dots, b_n$ расположены вдоль главной диагонали; c_1, $c_2, \dots, c_{n-1}$ — над ней; $a_2, a_3, \dots, a_n$ — под ней.

Для нахождения собственных значений нужно приравнять нулю определитель $D_n(\lambda) = \det(A - \lambda E)$, или

$$D_n(\lambda) = \begin{vmatrix} b_1 - \lambda & c_1 & & & & 0 \\ a_2 & b_2 - \lambda & c_2 & & & \\ \dots & \dots & \dots & \dots & \dots & \dots \\ & & & a_{n-1} & b_{n-1} - \lambda & c_{n-1} \\ 0 & & & & a_n & b_n - \lambda \end{vmatrix} = 0. \tag{4.47}$$

Произвольный определитель n-го порядка можно выразить через n миноров $(n-1)$-го порядка путем разложения его по элементам любой строки или любого столбца. Разложим определитель (4.47) по элементам последней строки, в которой всего два ненулевых элемента. Получим

$$D_n(\lambda) = (b_n - \lambda) D_{n-1}(\lambda) - a_n M_{n-1}(\lambda), \tag{4.48}$$

$$M_{n-1}(\lambda) = \begin{vmatrix} b_1 - \lambda & c_1 & & & 0 \\ a_2 & b_2 - \lambda & c_2 & & \\ \dots & \dots & \dots & \dots & \dots \\ & 0 & & a_{n-1} & c_{n-1} \end{vmatrix}.$$

Поскольку минор $M_{n-1}(\lambda)$ содержит в последнем столбце лишь один нелулевой элемент c_{n-1}, то, разлагая его по элементам этого столбца, получаем

$$M_{n-1}(\lambda) = c_{n-1} D_{n-2}(\lambda).$$

Подставляя это выражение в формулу (4.48), получаем рекуррентные соотношения, выражающие минор высшего порядка через миноры двух низших порядков:

$$D_n(\lambda) = (b_n - \lambda) D_{n-1}(\lambda) - a_n c_{n-1} D_{n-2}(\lambda). \tag{4.49}$$

Положим $D_0(\lambda) = 1$. Минор первого порядка равен элементу a_{11} определителя, т. е. в данном случае $D_1(\lambda) = b_1 - \lambda$. Проверим, с учетом значений $D_0(\lambda)$, $D_1(\lambda)$ правильность формулы (4.49) при $n = 2$:

$$D_2(\lambda) = (b_2 - \lambda)D_1(\lambda) - a_2 c_1 D_0(\lambda) = (b_2 - \lambda)(b_1 - \lambda) - a_2 c_1. \quad (4.50)$$

Вычисляя минор второго порядка определителя (4.47), убеждаемся в справедливости выражения (4.50). Таким образом, используя рекуррентные соотношения (4.49), можно найти выражение для характеристического многочлена $D_n(\lambda)$ для любого $n \geqslant 2$. Вычисляя корни этого многочлена, получаем собственные значения $\lambda_1, \lambda_2, \ldots, \lambda_n$ трехдиагональной матрицы (4.46).

Будем считать, что собственные значения $\lambda_1, \lambda_2, \ldots, \lambda_n$ матрицы (4.46) вычислены. Найдем соответствующие им собственные векторы. Для любого собственного значения собственный вектор находится из системы уравнений (4.36)

$$(A - \lambda E)\mathbf{x} = \mathbf{0}.$$

Перейдем от матричной формы записи этой системы к развернутой (A — матрица вида (4.46), $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$):

$$\begin{aligned}
&(b_1 - \lambda)x_1 + c_1 x_2 &&= 0,\\
&a_2 x_1 + (b_2 - \lambda)x_2 + c_2 x_3 &&= 0,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{n-1}x_{n-2} + (b_{n-1} - \lambda)x_{n-1} + c_{n-1}x_n &&= 0,\\
&a_n x_{n-1} + (b_n - \lambda)x_n &&= 0.
\end{aligned} \quad (4.51)$$

Матрица системы (4.51) вырожденная, поскольку ее определитель (4.47) равен нулю. Можно показать, что если $c_i \neq 0$ $(i = 1, 2, \ldots, n-1)$, то последнее уравнение системы (4.51) является следствием остальных уравнений. Действительно, если отбросить первый столбец и последнюю строку в матрице A, то вместо (4.47) получится определитель вида

$$\begin{vmatrix}
c_1 & & & 0\\
b_2 - \lambda & c_2 & & \\
\ldots & \ldots & \ldots & \ldots\\
0 & a_{n-1} & b_{n-1} - \lambda & c_{n-1}
\end{vmatrix} = c_1 c_2 \ldots c_{n-1} \neq 0. \quad (4.52)$$

Следовательно, все строки с первой по $(n-1)$-ю линейно независимы, последнее уравнение системы (4.51) — следствие остальных, и одно неизвестное этой системы является свободным. Отбрасывая последнее

уравнение системы (4.51), записываем ее в виде

$$
\begin{array}{ll}
c_1x_2 & = -(b_1-\lambda)x_1, \\
(b_2-\lambda)x_2 + c_2x_3 & = -a_2x_1, \\
\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots & \\
\qquad a_{n-1}x_{n-2} + (b_{n-1}-\lambda)x_{n-1} + c_{n-1}x_n = 0. &
\end{array}
\tag{4.53}
$$

Считая компоненту x_1 свободным неизвестным и полагая ее равной любому ненулевому значению (иначе при $x_1 = 0$ остальные компоненты также будут нулевыми), можно из системы (4.53) найти последовательно все остальные компоненты: из первого уравнения легко вычислить x_2, из второго x_3 и т. д., из последнего x_n. Поскольку определитель (4.52) этой системы отличен от нуля, то она имеет единственное решение. Описанным способом могут быть найдены собственные векторы, соответствующие всем собственным значениям $\lambda_1, \lambda_2, \dots, \lambda_n$ трехдиагональной матрицы (4.46).

Если трехдиагодальная матрица получена в результате последовательности преобразований подобия исходной симметричной матрицы, то, как уже отмечалось, все собственные значения трехдиагональной матрицы являются одновременно собственными значениями исходной матрицы, а собственные векторы пересчитываются по формулам (4.45). При этом вычислять произведения матриц $P_{n-1,n}, \dots, P_{24}, P_{23}$ на собственные векторы трехдиагональной матрицы нецелесообразно, поскольку при умножении матрицы P_{ij} на вектор $\mathbf{x}$ изменяются только две его компоненты: x_i и x_j. Поэтому в качестве этих компонент берем значения $px_i - qx_j$ и $qx_i + px_j$, что сокращает объем вычислений по сравнению с умножением матрицы P_{ij} на вектор $\mathbf{x}$.

4. Частичная проблема собственных значений. Часто в практических вычислениях бывают нужны не все собственные значения, а лишь некоторые из них. В этих случаях нецелесообразно решать полную проблему собственных значений.

Для решения *частичной проблемы собственных значений*, состоящей в определении одного или нескольких собственных значений и соответствующих им собственных векторов, обычно используются итерационные методы. Строится такой итерационный процесс, который сходится к одному собственному значению и собственному вектору, причем используемые алгоритмы обычно весьма экономичны.

Построим итерационный процесс, применяя метод итераций к решению системы уравнений

$$\lambda\mathbf{x} = A\mathbf{x}. \tag{4.54}$$

Модифицируем рассмотренный ранее метод простой итерации (4.26).

Запишем (4.54), введя вспомогательный вектор $\mathbf{y}$:

$$\mathbf{y} = A\mathbf{x}, \tag{4.55}$$

$$\lambda\mathbf{x} = \mathbf{y}. \tag{4.56}$$

Пусть $\mathbf{x}^{(0)}$ — начальное приближение собственного вектора $\mathbf{x}$, причем собственные векторы на каждой итерации нормированы, так что $|\mathbf{x}^{(k)}| = 1$ ($k = 0, 1, \dots$). Используем соотношение (4.55) для вычисления $\mathbf{y}^{(1)}$:

$$\mathbf{y}^{(1)} = A\mathbf{x}^{(0)}.$$

Соотношение (4.56) используем для вычисления первого приближения $\lambda^{(1)}$, применяя умножение обеих частей равенства скалярно на $\mathbf{x}^{(0)}$:

$$\lambda^{(1)}\mathbf{x}^{(0)} = \mathbf{y}^{(1)}, \quad \lambda^{(1)}\mathbf{x}^{(0)} \cdot \mathbf{x}^{(0)} = \mathbf{y}^{(1)} \cdot \mathbf{x}^{(0)},$$

$$\lambda^{(1)} = \frac{\mathbf{y}^{(1)} \cdot \mathbf{x}^{(0)}}{\mathbf{x}^{(0)} \cdot \mathbf{x}^{(0)}} = \mathbf{y}^{(1)} \cdot \mathbf{x}^{(0)}.$$

Здесь учтено, что вектор $\mathbf{x}^{(0)}$ нормирован, т. е. $\mathbf{x}^{(0)} \cdot \mathbf{x}^{(0)} = 1$. Следующее приближение собственного вектора $\mathbf{x}^{(1)}$ можно вычислить, нормируя вектор $\mathbf{y}^{(1)}$.

Окончательно итерационный процесс записывается в виде

$$\begin{aligned} \mathbf{y}^{(k+1)} &= A\mathbf{x}^{(k)}, \\ \lambda^{(k+1)} &= \mathbf{y}^{(k+1)} \cdot \mathbf{x}^{(k)}, \\ \mathbf{x}^{(k+1)} &= \frac{\mathbf{y}^{(k+1)}}{|\mathbf{y}^{(k+1)}|}, \quad k = 0, 1, \dots, \end{aligned} \tag{4.57}$$

и продолжается до установления постоянных значений λ и $\mathbf{x}$. При этом нужно учесть, что применяя критерии завершения итераций, следует проверять близость векторов $(\operatorname{sign}\lambda^{(k+1)})\mathbf{x}^{(k+1)}$ и $\mathbf{x}^{(k)}$, а не $\mathbf{x}^{(k+1)}$ и $\mathbf{x}^{(k)}$.

Можно показать, что найденное в результате итерационного процесса (4.57) число λ является наибольшим по модулю собственным значением данной матрицы A, а $\mathbf{x}$ — соответствующим ему собственным вектором. Скорость сходимости этого итерационного процесса зависит от удачного выбора начального приближения. Если начальный вектор близок к истинному собственному вектору, то итерации сходятся быстро.

Для решения системы (4.54) можно использовать и другие итерационные методы. В частности, метод Ньютона (см. гл. 5, § 3) дает лучшую сходимость, если удачно выбрано начальное приближение $\mathbf{x}^{(0)}$. В этом случае бывает достаточно нескольких итераций.

В некоторых задачах нужно искать не наибольшее, а наименьшее по модулю собственное значение матрицы A. В этом случае можно умножить систему (4.54) на обратную матрицу A^{-1}:

$$\lambda A^{-1}\mathbf{x} = A^{-1}A\mathbf{x}.$$

Отсюда, деля обе части этого равенства на λ и учитывая, что $A^{-1}A = E$, получаем

$$\frac{1}{\lambda}\,\mathbf{x} = A^{-1}\mathbf{x}. \tag{4.58}$$

Следовательно, $1/\lambda$ является собственным значения обратной матрицы, и задача (4.58) отличается от ранее рассматриваемой тем, что здесь будет вычисляться наибольшее по модулю собственное значение $1/\lambda$ матрицы A^{-1}, что будет достигнуто при наименьшем по модулю λ. Заметим также, что процесс (4.57) может быть использован для нахождения наименьшего по модулю собственного значения обратной матрицы.

Упражнения

1. Провести геометрический анализ единственности решения системы трех линейных уравнений с тремя неизвестными в зависимости от значения определителя.

2. Элементы треугольной матрицы вводятся построчно в память машины. Записать алгоритм вычисления определителя данной матрицы.

3. Используя метод Гаусса, решить следующие системы уравнений с погрешностью 10^{-4}:

а) $$\begin{aligned} 1.17\,x_1 + 0.53\,x_2 - 0.84\,x_3 &= 1.15,\\ 0.64\,x_1 - 0.72\,x_2 - 0.43\,x_3 &= 0.15,\\ 0.32\,x_1 + 0.43\,x_2 - 0.93\,x_3 &= -0.48; \end{aligned}$$

б) $$\begin{aligned} 1.20\,x_1 - 0.20\,x_2 + 0.30\,x_3 &= -0.60,\\ -0.20\,x_1 + 1.60\,x_2 - 0.10\,x_3 &= 0.30,\\ -0.30\,x_1 + 0.10\,x_2 - 1.50\,x_3 &= 0.40. \end{aligned}$$

4. Опытным путем оценить максимальный порядок системы уравнений, поддающейся решению методом Гаусса на доступном компьютере. Принять во внимание затраты оперативной памяти, времени счета, а также погрешность решения.

5. Записать алгоритм вычисления обратной матрицы по методу Гаусса:
а) воспользоваться готовым алгоритмом метода Гаусса (см. рис. 4.2);
б) для уменьшения числа арифметических действий модифицировать блок прямого хода данного алгоритма, так чтобы при исключении неизвестных обрабатывались сразу все правые части.

6. С помощью метода прогонки решить систему уравнений

$$\begin{aligned} 2x_1 + 2x_2 &= 1,\\ -x_1 + x_2 - 0.5x_3 &= 0,\\ x_2 - 3x_3 - x_4 &= 2,\\ x_3 + 2x_4 &= 2. \end{aligned}$$

7. Решить методом Гаусса–Зейделя с погрешностью 10^{-3} системы уравнений:

а) $\begin{aligned} 5.6x_1 + 2.7x_2 - 1.7x_3 &= 1.9, \\ 3.4x_1 - 3.6x_2 - 6.7x_3 &= -2.4, \\ 0.8x_1 + 1.3x_2 + 3.7x_3 &= 1.2; \end{aligned}$

б) $\begin{aligned} 7.1x_1 + 6.8x_2 + 6.1x_3 &= 7.0, \\ 5.0x_1 + 4.8x_2 + 5.3x_3 &= 6.1, \\ 8.2x_1 + 7.8x_2 + 7.1x_3 &= 5.8. \end{aligned}$

8. Выполнить упр. 4 для метода Гаусса–Зейделя.

9. Найти собственные значения и собственные векторы матриц

$$A = \begin{pmatrix} 3 & -1 \\ 2 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 2 \\ 0 & 2 & 3 \end{pmatrix}.$$

10. Записать алгоритм приведения матрицы четвертого порядка к трехдиагональному виду и решения полной проблемы собственных значений.

11*. Доказать утверждения, использовавшиеся в п. 2 § 4:
а) $P_{ij}^{-1} = P_{ij}^{\mathrm{T}}$;
б) в результате элементарного вращения симметрическая матрица преобразуется в симметрическую;
в) нулевые элементы, полученные в прямом методе вращений, не изменяются на последующих шагах метода.

12. Записать алгоритм вычисления наибольшего собственного значения с помощью итерационного метода.

ГЛАВА 5

НЕЛИНЕЙНЫЕ УРАВНЕНИЯ

§ 1. Уравнения с одним неизвестным

1. Вводные замечания. Задача нахождения корней нелинейных уравнений вида

$$F(x) = 0 \tag{5.1}$$

встречается в различных областях научных исследований (здесь $F(x)$ — некоторая непрерывная функция). Нелинейные уравнения можно разделить на два класса — алгебраические и трансцендентные. *Алгебраическими* уравнениями называются уравнения, содержащие только алгебраические функции (целые, рациональные, иррациональные). В частности, многочлен является целой алгебраической функцией. Уравнения, содержащие другие функции (тригонометрические, показательные, логарифмические и др.), называются *трансцендентными*.

Методы решения нелинейных уравнений делятся на прямые и итерационные. *Прямые* методы позволяют записать корни в виде некоторого конечного соотношения (формулы). Из школьного курса алгебры читателю известны такие методы для решения тригонометрических, логарифмических, показательных, а также простейших алгебраических уравнений.

Однако встречающиеся на практике уравнения не удается решить такими простыми методами. Для их решения используются *итерационные* методы, т. е. методы последовательных приближений. Как и в рассмотренном в гл. 4 случае систем линейных уравнений, алгоритм нахождения корня нелинейного уравнения с помощью итерационного метода состоит из двух этапов: а) отыскания приближенного значения корня (начального приближения); б) уточнения приближенного значения до некоторой заданной степени точности. В некоторых методах отыскивается не начальное приближение, а некоторый отрезок, содержащий корень.

Начальное приближение может быть найдено различными способами: из физических соображений, из решения аналогичной задачи при других исходных данных, с помощью графических методов. Если такие априорные оценки исходного приближения провести не удается, то находят две близко расположенные точки a и b, в которых непрерывная функция $F(x)$ принимает значения разных знаков, т. е. $F(a)F(b) < 0$. В этом случае между точками a и b есть по крайней мере одна точка, в которой $F(x) = 0$. В качестве начального приближения x_0 можно принять середину отрезка $[a, b]$, т. е. $x_0 = (a + b)/2$.

Общая схема итерационных методов была изложена в § 3 гл. 4 (см. рис. 4.5). Напомним, что итерационный процесс состоит в последовательном уточнении начального приближения x_0. Каждый такой *шаг* называется *итерацией*. В результате итераций находится последовательность

приближенных значений корня x_1, x_2, ..., x_k, ... Если эти значения с ростом k стремятся к истинному значению корня

$$\lim_{k\to\infty} x_k = x,$$

то говорят, что итерационный процесс *сходится* [1].

Ниже рассматриваются некоторые итерационные методы решения трансцендентных уравнений. Они могут использоваться также и для нахождения корней алгебраических уравнений, особенности решения которых будут рассмотрены в § 2.

2. Метод деления отрезка пополам (метод бисекции). Это один из простейших методов нахождения корней нелинейных уравнений. Он состоит в следующем. Допустим, что нам удалось найти отрезок $[a, b]$, на котором расположено искомое значение корня [2] $x = c$, т. е. $c \in [a, b]$. В качестве начального приближения корня c_0 принимаем середину этого отрезка: $c_0 = (a + b)/2$. Далее исследуем значения функции $F(x)$ на концах отрезков $[a, c_0]$ и $[c_0, b]$, т. е. в точках a, c_0, b. Тот из отрезков, на концах которого $F(x)$ принимает значения разных знаков, содержит искомый корень; поэтому его принимаем в качестве нового отрезка $[a_1, b_1]$. Вторую половину отрезка $[a, b]$, на которой знак $F(x)$ не меняется, отбрасываем. В качестве первого приближения корня принимаем середину нового отрезка $c_1 = (a_1 + b_1)/2$ и т. д. Таким образом, k-е приближение вычисляется как

$$c_k = \frac{a_k + b_k}{2}\,; \tag{5.2}$$

после каждой итерации отрезок, на котором расположен корень, уменьшается вдвое, а после k итераций он сокращается в 2^k раз:

$$b_k - a_k = \frac{b - a}{2^k}\,. \tag{5.3}$$

Пусть приближенное решение x_* требуется найти с точностью до некоторого заданного малого числа $\varepsilon > 0$:

$$|x - x_*| < \varepsilon. \tag{5.4}$$

Взяв в качестве приближенного решения k-е приближение корня: $x_* = c_k$, запишем (5.4) с учетом обозначения $x = c$ в виде

$$|c - c_k| < \varepsilon. \tag{5.5}$$

---

[1] В отличие от гл. 4 в (5.1) x является скаляром, а не вектором. Поэтому номер итерации будем для простоты обозначать нижним индексом.

[2] Далее мы предполагаем, что $x = c$ — *единственный* корень на отрезке $[a, b]$. Если корней на $[a, b]$ несколько, то в результате применения метода деления отрезка пополам и метода хорд (см. п. 3) будет найдено приближенное значение *одного* из корней.

Как легко видеть, из (5.2) следует, что (5.5) выполнено, если

$$b_k - a_k < 2\varepsilon. \tag{5.6}$$

Таким образом, итерационный процесс нужно продолжать до тех пор, пока не будет выполнено условие (5.6).

Метод деления отрезка пополам проиллюстрирован на рис. 5.1. Пусть для определенности $F(a) < 0$, $F(b) > 0$. В качестве начального приближения корня примем $c_0 = (a + b)/2$. Поскольку в рассматриваемом случае $F(c_0) < 0$, то $c \in [c_0, b]$, и рассматриваем только отрезок $[c_0, b]$, т. е. $a_1 = c_0$, $b_1 = b$. Следующее приближение: $c_1 = (c_0 + b)/2$. При этом отрезок $[c_1, b]$ отбрасываем, поскольку $F(c_1) > 0$ и $F(b) > 0$. Таким образом, $c \in [c_0, c_1]$, $a_2 = c_0$, $b_2 = c_1$. Аналогично находим другие приближения: $c_2 = (c_0 + c_1)/2$ и т. д. до выполнения условия (5.6).

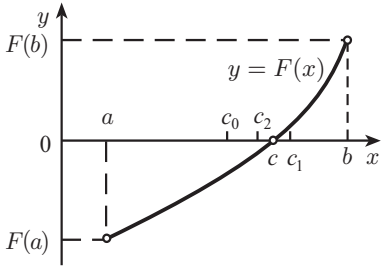


Рис. 5.1. Метод деления отрезка пополам

В отличие от большинства других итерационных методов метод деления отрезка пополам всегда сходится, причем можно гарантировать, что полученное решение будет иметь любую наперед заданную точность (разумеется, в рамках разрядности компьютера). При применении этого метода нет необходимости приближенно определять момент достижения требуемой точности, пользуясь, например, условиями близости двух последовательных приближений (4.22) или (4.23) (записанными для скалярного случая). Вместо них применяется соотношение (5.6), гарантирующее выполнение (5.4).

Однако метод деления отрезка пополам довольно медленный. Вычислим число итераций N, требуемое для достижения точности ε. Для этого выясним, пользуясь (5.3), для каких k выполнено условие (5.6), и возьмем в качестве N наименьшее из таких k. Окончательно получим

$$k > \log_2 \frac{b-a}{2\varepsilon}, \quad N = E\left(\log_2 \frac{b-a}{2\varepsilon}\right) + 1, \tag{5.7}$$

где $E(x)$ — целая часть числа x. Обычно для метода деления отрезка пополам N больше, чем для некоторых других методов, что не является препятствием к применению этого метода, если каждое вычисление значения функции $F(x)$ несложно.

Итерационный процесс можно завершать и тогда, когда значение функции $F(x)$ после k-й итерации станет меньшим по модулю ε, т. е.

$$|F(c_k)| < \varepsilon. \tag{5.8}$$

Такое условие окончания итераций аналогично условию (4.24). Действительно, для уравнения (5.1) величина $F(c_k)$ есть *невязка*, полученная на k-й итерации.

На рис. 5.2 представлен алгоритм итерационного процесса нахождения корня уравнения (5.1) методом деления отрезка пополам. Здесь сужение отрезка производится путем замены границ a или b на текущее значение корня c. При этом значение $F(a)$ вычисляется лишь один раз, поскольку нам нужен только знак функции $F(x)$ на левой границе, а он в процессе итераций не меняется.

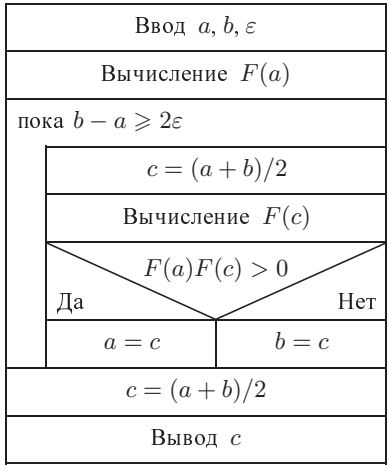


Рис. 5.2. Алгоритм метода деления отрезка пополам

3. Метод хорд. Пусть мы нашли отрезок $[a, b]$, на котором функция $F(x)$ меняет знак. Для определенности примем $F(a) > 0$, $F(b) < 0$ (рис. 5.3). В данном методе процесс итераций состоит в том, что в качестве приближений корню уравнения (5.1) принимаются значения $c_0, c_1, \ldots$ точек пересечения хорды с осью абсцисс.

Сначала находим уравнение хорды AB:

$$\frac{y - F(a)}{F(b) - F(a)} = \frac{x - a}{b - a}\,.$$

Для точки пересечения ее с осью абсцисс ($x = c_0$, $y = 0$) получим уравнение

$$c_0 = a - \frac{b - a}{F(b) - F(a)} F(a). \qquad (5.9)$$

Далее, сравнивая знаки величин $F(a)$ и $F(c_0)$ для рассматриваемого случая, приходим к выводу, что корень находится в интервале (a, c_0), так как $F(a)F(c_0) < 0$. Отрезок $[c_0, b]$ отбрасываем. Следующая итерация состоит в определении нового приближения c_1 как точки пересечения хорды AB_1 с осью абсцисс и т. д.

В отличие от метода деления отрезка пополам в методе хорд условие окончания итераций типа (5.6) неприменимо. Так, на рис. 5.3 видно, что длина отрезка $[a, c_k]$ никогда не станет меньше длины отрезка $[a, c]$. Вместо (5.6) нужно использовать условие близости двух последовательных приближений

$$|c_k - c_{k-1}| < \varepsilon \qquad (5.10)$$

или условием малости невязки (5.8).

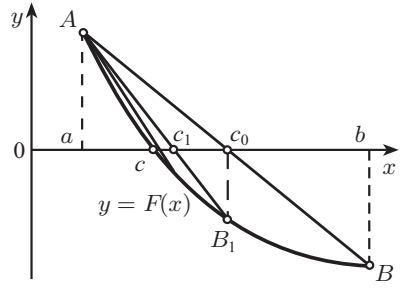


Рис. 5.3. Метод хорд

Метод деления отрезка пополам и метод хорд весьма похожи, в частности, процедурой проверки знаков функции на концах отрезка. При этом

второй из них в ряде случаев дает более быструю сходимость итерационного процесса. Кроме того, оба рассмотренных метода не требуют знания дополнительной информации о функции $F(x)$. Например, не требуется, чтобы функция была дифференцируема. Непрерывность $F(x)$ гарантирует успех применения этих методов. Более сложные методы решения нелинейных уравнений используют дополнительную информацию о функции $F(x)$, прежде всего свойство дифференцируемости функции. Как результат они обычно обладают более быстрой сходимостью, но в то же время, применимы для более узкого класса функций, и их сходимость не всегда гарантирована. Примером такого метода служит метод Ньютона.

4. Метод Ньютона (метод касательных). Его отличие от предыдущего метода состоит в том, что на k–й итерации вместо хорды проводится касательная к кривой $y = F(x)$ при $x = c_{k-1}$ и ищется точка пересечения касательной с осью абсцисс. При этом не обязательно задавать отрезок $[a, b]$, содержащий корень уравнения (5.1), а достаточно лишь найти некоторое начальное приближение корня $x = c_0$ (рис. 5.4).

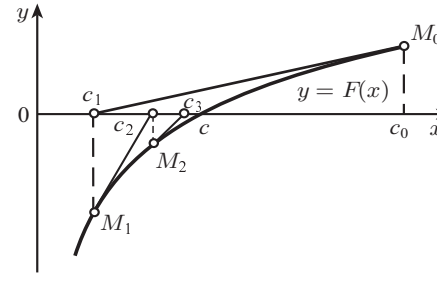


Рис. 5.4. Метод касательных

Уравнение касательной, проведенной к кривой $y = F(x)$ в точке M_0 с координатами c_0 и $F(c_0)$, имеет вид

$$y - F(c_0) = F'(c_0)(x - c_0).$$

Отсюда найдем следующее приближение корня c_1 как абсциссу точки пересечения касательной с осью x ($y = 0$):

$$c_1 = c_0 - F(c_0)/F'(c_0).$$

Аналогично могут быть найдены и следующие приближения как точки пересечения с осью абсцисс касательных, проведенных в точках M_1, M_2 и т. д. Формула для k-го приближения имеет вид

$$c_k = c_{k-1} - F(c_{k-1})/F'(c_{k-1}), \quad k = 1, 2, \ldots \tag{5.11}$$

При этом необходимо, чтобы $F'(c_{k-1})$ не равнялась нулю. Для окончания итерационного процесса могут быть использованы условия (5.10) или (5.8).

Из (5.11) следует, что на каждой итерации объем вычислений в методе Ньютона больший, чем в рассмотренных ранее методах, поскольку приходится находить значение не только функции $F(x)$, но и ее производной. Однако скорость сходимости здесь значительно выше, чем в других методах.

Остановимся на некоторых вопросах, связанных со сходимостью метода Ньютона и его использованием. Имеет место следующая теорема.

Т е о р е м а. *Пусть* $x = c$ *— корень уравнения* (5.1), *т. е.* $F(c) = 0$, *а* $F'(c) \neq 0$ *и* $F''(x)$ *непрерывна. Тогда существует окрестность* D *корня* c ($c \in D$) *такая, что если начальное приближение* c_0 *принадлежит этой окрестности, то для метода Ньютона последовательность значений* $\{c_k\}$ *сходится к* c *при* $k \to \infty$. *При этом для погрешности корня* $\varepsilon_k = c - c_k$ *имеет место соотношение*

$$\lim_{k\to\infty} \left| \frac{\varepsilon_k}{\varepsilon_{k-1}^2} \right| = \left| \frac{F''(c)}{2F'(c)} \right|.$$

Фактически это означает, что на каждой итерации погрешность возводится в квадрат, т. е. число верных знаков корня удваивается. Если

$$\left| \frac{F''(c)}{2F'(c)} \right| \sim 1,$$

то легко показать, что при $|\varepsilon_0| \leqslant 0.5$ пяти-шести итераций достаточно для получения минимально возможной погрешности при вычислениях с двойной точностью. Действительно, погрешность теоретически станет в этом случае величиной порядка 2^{-64}, что намного меньше, чем максимальная погрешность округления при вычислениях с двойной точностью, равная 2^{-53} (см. гл. 1, § 2). Заметим, что для получения столь малой погрешности в методе деления отрезка пополам потребовалось бы согласно (5.7) более 50 итераций.

П р и м е р. Для иллюстрации рассмотрим уравнение $x^2 - 0.25 = 0$ и найдем методом Ньютона один из его корней, например $x = c = 0.5$. Для данного уравнения $F''(c)/2F'(c) = 1$. Выберем $c_0 = 1$, тогда $\varepsilon_0 = -0.5$. Проводя вычисления с двойной точностью, получим следующие значения погрешностей:

$$\varepsilon_1 = -1.25 \cdot 10^{-1}, \qquad \varepsilon_3 = -1.52 \cdot 10^{-4}, \qquad \varepsilon_5 = -5.55 \cdot 10^{-16},$$
$$\varepsilon_2 = -1.25 \cdot 10^{-2}, \qquad \varepsilon_4 = -2.32 \cdot 10^{-8}, \qquad \varepsilon_6 = 0.$$

Таким образом, после шести итераций погрешность в рамках арифметики с двойной точностью исчезла.

Трудность в применении метода Ньютона состоит в выборе начального приближения, которое должно находиться в окрестности D. При неудачном выборе начального приближения итерации могут расходиться.

П р и м е р. Для уравнения $\operatorname{arctg} x = 0$ (корень $x = c = 0$) при начальном приближении $c_0 = 1.5$ первые шесть итераций приводят к погрешностям

$$\varepsilon_1 = \quad 1.69, \qquad \varepsilon_3 = \quad 5.11, \qquad \varepsilon_5 = \quad 1.58 \cdot 10^3,$$
$$\varepsilon_2 = -2.32, \qquad \varepsilon_4 = -32.3, \qquad \varepsilon_6 = -3.89 \cdot 10^6.$$

Очевидно, что итерации здесь расходятся.

Для предотвращения расходимости иногда целесообразно использовать смешанный алгоритм. Он состоит в том, что сначала применяется

всегда сходящийся метод (например, метод деления отрезка пополам), а после некоторого числа итераций — быстро сходящийся метод Ньютона.

5. Метод простой итерации. Для использования этого метода исходное нелинейное уравнение записывается в виде

$$x = f(x). \tag{5.12}$$

Пусть известно начальное приближение корня $x = c_0$. Подставляя это значение в правую часть уравнения (5.12), получаем новое приближение

$$c_1 = f(c_0).$$

Подставляя каждый раз новое значение корня в (5.12), получаем последовательность значений

$$c_k = f(c_{k-1}), \quad k = 1, 2, \ldots$$

Итерационный процесс прекращается, если результаты двух последовательных итераций близки, т. е. если выполнено неравенство (5.10). Заметим, что в методе простой итерации для невязки, полученной на k-й итерации, выполнено соотношение

$$r_k = c_k - f(c_k) = c_k - c_{k+1}.$$

Таким образом, условие малости невязки на k-й итерации оказывается эквивалентным условию близости k-го и $k+1$-го приближений.

Достаточное условие сходимости метода простой итерации дается следующей теоремой.

Т е о р е м а. *Пусть $x = c$ — корень уравнения* (5.12), *т. е. $c = f(c)$, а $|f'(c)| < 1$ и $f'(x)$ непрерывна. Тогда существует окрестность D корня c ($c \in D$) такая, что если начальное приближение c_0 принадлежит этой окрестности, то для метода простой итерации последовательность значений $\{c_k\}$ сходится к c при $k \to \infty$.*

Метод простой итерации рассмотрен нами для уравнения (5.12). К такому виду можно привести и более общее уравнение (5.1), аналогично тому, как это делалось при решении систем линейных уравнений:

$$F(x) = 0, \quad \tau F(x) = 0,$$
$$x = x - \tau F(x). \tag{5.13}$$

Здесь $\tau \neq 0$ — некоторое число. Уравнение (5.13) эквивалентно (5.12) с функцией $f(x) = x - \tau F(x)$. За счет выбора значения параметра τ можно добиваться сходимости метода простой итерации и повышения скорости сходимости. Например, если на некотором отрезке, содержащем корень уравнения, производная $F'(x)$ ограничена константами m и M:

$$0 < m < F'(x) < M,$$

то для производной $f'(x)$ будет справедливо неравенство

$$1 - \tau M < f'(x) < 1 - \tau m.$$

Выбирая $\tau = 2/(M+m)$, получаем

$$-\frac{M-m}{M+m} < f'(x) < \frac{M-m}{M+m},$$

т. е. $|f'(x)| < 1$, что обеспечивает сходимость метода простой итерации.

Параметр τ в (5.13) можно выбирать и переменным, зависящим от номера итерации. Так, если положить $\tau_k = 1/F'(c_{k-1})$, то метод простой итерации для уравнения (5.13) примет вид

$$c_k = c_{k-1} - F(c_{k-1})/F'(c_{k-1}).$$

Это соотношение совпадает с формулой метода Ньютона (5.11). Следовательно, метод Ньютона можно трактовать как частный случай метода простой итерации с переменным τ.

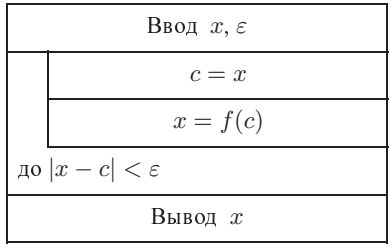


Рис. 5.5. Алгоритм метода простой итерации

На рис. 5.5 представлен алгоритм решения нелинейного уравнения (5.12) методом простой итерации. Здесь x — начальное приближение корня, а в дальнейшем — значение корня после каждой итерации, c — результат предыдущей итерации. В данном алгоритме предполагалось, что итерационный процесс сходится. Если такой уверенности нет, то необходимо ограничить число итераций и ввести для них счетчик (см. рис. 4.6).

§ 2. О решении алгебраических уравнений

1. Действительные корни. Рассмотренные выше методы решения нелинейных уравнений пригодны как для трансцендентных, так и для алгебраических уравнений. Вместе с тем при нахождении корней многочленов приходится сталкиваться с некоторыми особенностями. В частности, при рассмотрении точности вычислительного процесса (гл. 1, § 3) отмечалась чувствительность к погрешностям значений корней многочлена. С другой стороны, по сравнению с трансцендентными функциями многочлены имеют то преимущество, что заранее известно число их корней.

Напомним некоторые известные из курса алгебры свойства алгебраических уравнений с действительными коэффициентами вида

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0. \tag{5.14}$$

1. Уравнение степени n имеет всего n корней с учетом кратности, среди которых могут быть как действительные, так и комплексные.

2. Комплексные корни образуют комплексно–сопряженные пары, т. е. каждому корню $x = c + id$ соответствует корень $x = c - id$.

Одним из способов решения уравнения (5.14) является *метод понижения порядка*. Он состоит в том, что после нахождения какого – либо корня $x = c$ данное уравнение можно разделить на $x - c$, понизив его порядок до $n - 1$. Правда, при таком способе нужно помнить о точности, поскольку даже небольшая погрешность в значении первого корня может привести к накапливанию погрешности в дальнейших вычислениях.

Рассмотрим применение метода Ньютона к решению уравнения (5.14). В соответствии с формулой (5.11) итерационный процесс для нахождения корня нелинейного уравнения (5.14) имеет вид

$$x_k = x_{k-1} - \frac{F(x_{k-1})}{F'(x_{k-1})},$$

$$F(x) = a_0 + a_1 x + \ldots + a_n x^n, \quad F'(x) = a_1 + 2a_2 x + \ldots + n a_n x^{n-1}.$$

Для вычисления значений многочленов $F(x)$ и $F'(x)$ в точке $x = x_{k-1}$ может быть использована схема Горнера (см. гл. 2, § 1, п. 4).

Естественно, при использовании метода Ньютона должны выполняться условия сходимости (см. § 1, п. 4). При их соблюдении в результате численного решения получается значение того корня, который находится вблизи заданного начального приближения x_0.

Заметим, что для уменьшения погрешностей лучше сначала находить меньшие по модулю корни многочлена и сразу удалять их из уравнения, приводя его к меньшей степени. Поэтому, если отсутствует информация о величинах корней, в качестве начальных приближений принимают числа 0, ± 1 и т. д.

2. Комплексные корни. При использовании компьютера имеется возможность работать с комплексными числами; поэтому изложенный метод Ньютона может быть использован (с необходимым обобщением) и для нахождения комплексных корней многочленов. При этом, если в качестве начального приближения x_0 взять комплексное число, то последующие приближения и окончательное значение корня могут оказаться комплексными. Ниже рассмотрим другой подход к отысканию комплексных корней.

Комплексные корни попарно сопряженные, и при их исключении порядок уравнения уменьшается на два, поскольку оно делится сразу на квадратный трехчлен, т. е.

$$F(x) = (x^2 + px + q)(b_n x^{n-2} + \ldots + b_2) + b_1 x + b_0. \tag{5.15}$$

Линейный остаток $b_1 x + b_0$ равен нулю, если p, q выражаются с помощью найденных корней:

$$p = -2c, \quad q = c^2 + d^2, \quad x = c \pm id.$$

Представление (5.15) может быть также использовано для нахождения p, q, а значит, и для определения корней. Эта процедура лежит в основе *метода Лина*. Суть этого метода состоит в следующем. Предположим, что коэффициенты b_0, b_1 равны нулю. Тогда, сравнивая коэффициенты

при одинаковых степенях x многочлена $F(x)$ в выражениях (5.14) и (5.15), можно получить (для упрощения выкладок $b_n = a_n = 1$)

$$\begin{aligned} b_{n-1} &= a_{n-1} - p, \\ b_{n-2} &= a_{n-2} - pb_{n-1} - q, \\ &\dots\dots\dots\dots\dots \\ b_2 &= a_2 - pb_3 - qb_4; \end{aligned} \tag{5.16}$$

$$p = (a_1 - qb_3)/b_2, \quad q = a_0/b_2. \tag{5.17}$$

В соотношения (5.17) входят коэффициенты b_2 и b_3, которые являются функциями p и q. Действительно, задав значения p и q, из соотношений (5.16) можно последовательно найти b_3 и b_2. Поэтому соотношения (5.17) представляют собой систему двух нелинейных уравнений относительно p и q:

$$\begin{aligned} p &= f_1(p, q), \\ q &= f_2(p, q). \end{aligned}$$

Такая система в методе Лина решается методом простой итерации (см. п. 2 § 3): задаются начальные приближения для p, q которые используются для вычисления коэффициентов $b_{n-1}, b_{n-2}, \dots, b_2$, затем из уравнений (5.17) уточняются значения p, q. Итерационный процесс вычисления этих величин продолжается до тех пор, пока их изменения в двух последовательных итерациях не станут малыми.

Широко распространен также другой метод, основанный на выделении квадратичного множителя $x^2 + px + q$, — *метод Бэрстоу*. Он использует метод Ньютона для решения системы двух уравнений (см. п. 3 § 3).

§ 3. Системы уравнений

1. Вводные замечания. В гл. 4 рассматривались системы линейных уравнений. Многие практические задачи сводятся к решению *системы нелинейных уравнений*.

Пусть для вычисления неизвестных $x_1, x_2, \dots, x_n$ требуется решить систему n нелинейных уравнений

$$\begin{aligned} F_1(x_1, x_2, \dots, x_n) &= 0, \\ F_2(x_1, x_2, \dots, x_n) &= 0, \\ &\dots\dots\dots\dots\dots \\ F_n(x_1, x_2, \dots, x_n) &= 0. \end{aligned} \tag{5.18}$$

В векторной форме эту систему можно записать как

$$\mathbf{F}(\mathbf{x}) = \mathbf{0},$$

где

$$\mathbf{F} = \{F_1, F_2, \dots, F_n\}, \quad \mathbf{x} = \{x_1, x_2, \dots, x_n\}.$$

В отличие от систем линейных уравнений не существует прямых методов решения нелинейных систем общего вида. Лишь в отдельных случаях систему (5.18) можно решить непосредственно. Например, для случая двух уравнений иногда удается выразить одно неизвестное через другое и таким образом свести задачу к решению одного нелинейного уравнения относительно одного неизвестного.

Для решения систем нелинейных уравнений обычно используются итерационные методы. Ниже будут рассмотрены некоторые из них: метод простой итерации, метод Зейделя и метод Ньютона.

2. Метод простой итерации и метод Зейделя. Систему уравнений (5.18) представим в виде

$$\begin{aligned} x_1 &= f_1(x_1, x_2, \ldots, x_n), \\ x_2 &= f_2(x_1, x_2, \ldots, x_n), \\ &\ldots\ldots\ldots\ldots \\ x_n &= f_n(x_1, x_2, \ldots, x_n). \end{aligned} \tag{5.19}$$

Для решения этой системы можно использовать *метод простой итерации*, аналогичный соответствующему методу для одного уравнения. Значения неизвестных на k-й итерации будут найдены с использованием их значений на предыдущей итерации $x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_n^{(k-1)}$ как

$$x_i^{(k)} = f_i\big(x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_n^{(k-1)}\big), \quad i = 1, 2, \ldots, n. \tag{5.20}$$

Систему (5.19) можно решать и *методом Зейделя*, напоминающим метод Гаусса–Зейделя решения систем линейных уравнений (см. гл. 4, § 3). Значение $x_i^{(k)}$ находится из i-го уравнения системы (5.19) с использованием уже вычисленных на текущей итерации значений неизвестных. Таким образом, значения неизвестных на k-й итерации будут находиться не с помощью (5.20), а с помощью соотношения (ср. с (4.31))

$$x_i^{(k)} = f_i\big(x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i^{(k-1)}, \ldots, x_n^{(k-1)}\big), \quad i = 1, 2, \ldots, n.$$

Итерационный процесс в обоих методах продолжается до тех пор, пока изменения всех неизвестных в двух последовательных итерациях не станут малыми, т. е. в качестве критерия завершения итераций выбирается одно из условий (4.21) – (4.23).

При использовании метода простой итерации и метода Зейделя успех во многом определяется удачным выбором начальных приближений неизвестных: они должны быть достаточно близкими к истинному решению. В противном случае итерационный процесс может не сойтись.

3. Метод Ньютона. Этот метод обладает гораздо более быстрой сходимостью, чем метод простой итерации и метод Зейделя. В случае одного уравнения $F(x) = 0$ алгоритм метода Ньютона был легко получен путем записи уравнения касательной к кривой $y = F(x)$. По сути для нахождения нового приближения функция $F(x)$ заменялась линейной

функцией, т. е. раскладывалась в ряд Тейлора, при этом член, содержащий вторую производную, отбрасывался (как и все последующие члены). Та же идея лежит в основе метода Ньютона для системы уравнений: функции $F_i(x_1, x_2, \ldots, x_n)$ раскладываются в ряд Тейлора, причем в разложении отбрасываются члены, содержащие вторые (и более высоких порядков) производные.

Пусть приближенные значения неизвестных системы (5.18), полученные на предыдущей итерации, равны соответственно $x_1^{(k-1)}$, $x_2^{(k-1)}$, … … , $x_n^{(k-1)}$. Задача состоит в нахождении приращений (поправок) к этим значениям Δx_1, Δx_2, … , Δx_n, благодаря которым следующее приближение к решению системы (5.18) запишется в виде

$$\begin{aligned} x_1^{(k)} &= x_1^{(k-1)} + \Delta x_1, \\ x_2^{(k)} &= x_2^{(k-1)} + \Delta x_2, \\ x_n^{(k)} &= x_n^{(k-1)} + \Delta x_n. \end{aligned} \tag{5.21}$$

Проведем разложение левых частей уравнений (5.18) с учетом (3) в ряд Тейлора, ограничиваясь лишь линейными членами относительно приращений:

$$\begin{aligned} F_1\big(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\big) &\approx F_1 + \frac{\partial F_1}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial F_1}{\partial x_n}\Delta x_n, \\ F_2\big(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\big) &\approx F_2 + \frac{\partial F_2}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial F_2}{\partial x_n}\Delta x_n, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ F_n\big(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\big) &\approx F_n + \frac{\partial F_n}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial F_n}{\partial x_n}\Delta x_n. \end{aligned} \tag{5.22}$$

В правых частях этих соотношений значения F_1, F_2, … , F_n и их производных вычисляются в точке $\mathbf{x}^{(k-1)} = \big(x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_n^{(k-1)}\big)$.

Поскольку в соответствии с (5.18) левые части (5.22) должны обращаться в нуль, приравняем нулю и правые части, т. е. найдем новое приближение из условия равенства нулю разложений функций F_i. Получим следующую систему линейных алгебраических уравнений относительно приращений:

$$\begin{aligned} \frac{\partial F_1}{\partial x_1}\Delta x_1 + \frac{\partial F_1}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial F_1}{\partial x_n}\Delta x_n &= -F_1, \\ \frac{\partial F_2}{\partial x_1}\Delta x_1 + \frac{\partial F_2}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial F_2}{\partial x_n}\Delta x_n &= -F_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \frac{\partial F_n}{\partial x_1}\Delta x_1 + \frac{\partial F_n}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial F_n}{\partial x_n}\Delta x_n &= -F_n. \end{aligned} \tag{5.23}$$

Определителем системы (5.23) является *якобиан*

$$J = \begin{vmatrix} \dfrac{\partial F_1}{\partial x_1} & \dfrac{\partial F_1}{\partial x_2} & \cdots & \dfrac{\partial F_1}{\partial x_n} \\ \dfrac{\partial F_2}{\partial x_1} & \dfrac{\partial F_2}{\partial x_2} & \cdots & \dfrac{\partial F_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{\partial F_n}{\partial x_1} & \dfrac{\partial F_n}{\partial x_2} & \cdots & \dfrac{\partial F_n}{\partial x_n} \end{vmatrix}.$$

Для существования единственного решения системы (5.23) он должен быть отличным от нуля на каждой итерации.

Таким образом, итерационный процесс решения системы уравнений (5.18) методом Ньютона состоит в определении приращений Δx_1, Δx_2, $\ldots$, Δx_n к значениям неизвестных на каждой итерации посредством решения системы (5.23). Счет прекращается при выполнении одного из условий (4.21) – (4.23) или (4.24)[1]. Например, условие (4.22), которое с учетом (5.21) сведется к виду $\max\limits_{1\leqslant i\leqslant n} |\Delta x_i| < \varepsilon$. В методе Ньютона также важен удачный выбор начального приближения для обеспечения хорошей сходимости. Сходимость ухудшается с увеличением числа уравнений системы.

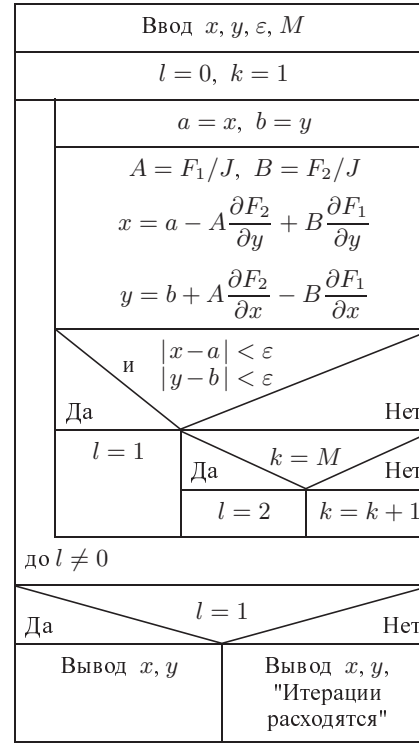


Рис. 5.6. Метод Ньютона для системы двух уравнений

В качестве примера рассмотрим использование метода Ньютона для решения системы двух уравнений

$$\begin{aligned} F_1(x, y) &= 0, \\ F_2(x, y) &= 0. \end{aligned} \qquad (5.24)$$

Пусть приближенные значения неизвестных равны a, b. Предположим, что якобиан системы (5.24)

[1] В этом условии невязка определяется как $\mathbf{r}^{(k)} = \mathbf{F}(\mathbf{x}^{(k)})$.

при $x = a,\ y = b$ отличен от нуля, т. е.

$$J = \begin{vmatrix} \dfrac{\partial F_1}{\partial x} & \dfrac{\partial F_1}{\partial y} \\ \dfrac{\partial F_2}{\partial x} & \dfrac{\partial F_2}{\partial y} \end{vmatrix} \neq 0.$$

Тогда следующие приближения неизвестных можно записать в виде

$$x = a - \frac{1}{J}\left(F_1 \frac{\partial F_2}{\partial y} - F_2 \frac{\partial F_1}{\partial y}\right),$$
$$y = b + \frac{1}{J}\left(F_1 \frac{\partial F_2}{\partial x} - F_2 \frac{\partial F_1}{\partial x}\right).$$

Величины, стоящие в правых частях, вычисляются при $x = a$, $y = b$.

Алгоритм метода Ньютона для решения системы двух уравнений изображен на рис. 5.6. В качестве исходных данных задаются начальные приближения неизвестных x, y, погрешность ε и допустимое число итераций M. В условной конструкции, проверяющей выполнение критерия завершения итераций, используется логическая операция «и», имеющаяся в современных языках программирования. Счетчик числа итераций организован так же, как и в алгоритме на рис. 4.6. Если итерации сойдутся, то выводятся значения x, y; в противном случае — текущие значения x, y и соответствующее сообщение.

Упражнения

1. Методом деления отрезка пополам найти с погрешностью 10^{-3} хотя бы один корень уравнений:
а) $2e^x = 5x$; б) $x^2 \cos 2x = -1$.

2. Записать алгоритм решения уравнения методом хорд.

3. Найти с погрешностью 10^{-3} методом хорд хотя бы один корень уравнений:
а) $2x - \lg x - 7 = 0$; б) $\operatorname{ctg} x - 0.1 = 0$.

4. Записать алгоритм решения уравнения методом Ньютона.

5. Используя метод Ньютона, найти с погрешностью 10^{-3} хотя бы один корень уравнений:
а) $\operatorname{tg}(0.55x + 0.1) = x^2$; б) $x^3 - 0.2x^2 + 0.5x + 1.5 = 0$.

6. Сравнить число итераций, необходимых для достижения заданной точности, в методах деления отрезка пополам, хорд и Ньютона на примере решения одного из уравнений, приведенных в упр. 3, 3, 3.

7*. Определить, при каких начальных приближениях метод Ньютона будет сходиться для уравнения $\operatorname{arctg} x = 0$.
Указание. Проиллюстрировать графически расходимость итераций в примере из § 1, п. 4. Получившееся в процессе выполнения упражнения уравнение решить численно одним из рассмотренных методов.

8. С помощью метода простой итерации найти с погрешностью 10^{-3} хотя бы один корень уравнений:
а) $5x - 8\ln x = 8$; б) $x^2 = \sin x$; в)* $xe^{2x} - 2 = 0$.

9. Определить глубину погружения деревянного шара радиуса 20 см, плавающего в воде. Плотность дерева 0.75 г/см^3.

10. Найти процентное содержание углекислого газа в реакции $2\mathrm{CO} + \mathrm{O}_2 \rightleftarrows 2\mathrm{CO}_2$, которое определяется уравнением $(p/k^2 - 1)x^3 + 3x - 2 = 0$, где p — давление, k — постоянная равновесия. Принять $p = 1$, $k = 1.648$.

11. Записать алгоритмы решения системы уравнений методом простой итерации и методом Зейделя.

12. Используя метод простой итерации и метод Зейделя, найти с погрешностью 10^{-3} хотя бы одно решение систем уравнений:

а) $x = y + \sin xy$,
$y = x + \cos(x + y)$;

б) $x - \operatorname{arctg} x - 0.2y \sin y + 1 = 0$,
$y - \operatorname{arctg} y - 0.3x \cos x - 1 = 0$.

ГЛАВА 6

МЕТОДЫ ОПТИМИЗАЦИИ

§ 1. Основные понятия

1. Определения. Под *оптимизацией* понимают процесс выбора наилучшего варианта из всех возможных. С точки зрения инженерных расчетов методы оптимизации позволяют выбрать наилучший вариант конструкции, наилучшее распределение ресурсов и т. п.

В процессе решения задачи оптимизации обычно необходимо найти оптимальные значения некоторых параметров, определяющих данную задачу. При решении инженерных задач их принято называть *проектными параметрами*, а в экономических задачах их обычно называют *параметрами плана*. В качестве проектных параметров могут быть, в частности, значения линейных размеров объекта, массы, температуры и т. п. Число n проектных параметров $x_1, x_2, \ldots, x_n$ характеризует размерность (и степень сложности) задачи оптимизации.

Выбор оптимального решения или сравнение двух альтернативных решений проводится с помощью некоторой зависимой величины (функции), определяемой проектными параметрами. Эта величина называется *целевой функцией* (или *критерием качества*). В процессе решения задачи оптимизации должны быть найдены такие значения проектных параметров, при которых целевая функция имеет минимум (или максимум). Таким образом, целевая функция — это глобальный критерий оптимальности в математических моделях, с помощью которых описываются инженерные или экономические задачи.

Целевую функцию можно записать в виде

$$u = f(x_1, x_2, \ldots, x_n). \tag{6.1}$$

Примерами целевой функции, встречающимися в инженерных и экономических расчетах, являются прочность или масса конструкции, мощность установки, объем выпуска продукции, стоимость перевозок грузов, прибыль и т. п.

В случае одного проектного параметра ($n = 1$) целевая функция (6.1) является функцией одной переменной, и ее график — некоторая кривая на плоскости. При $n = 2$ целевая функция является функцией двух переменных, и ее график — поверхность в трехмерном пространстве.

Следует отметить, что целевая функция не всегда может быть представлена в виде формулы. Иногда она может принимать только некоторые дискретные значения, задаваться в виде таблицы и т. п. Во всех случаях она должна быть однозначной функцией проектных параметров.

Целевых функций может быть несколько. Например, при проектировании изделий машиностроения одновременно требуется обеспечить

максимальную надежность, минимальную материалоемкость, максимальный полезный объем (или грузоподъемность). Некоторые целевые функции могут оказаться несовместимыми. В таких случаях необходимо вводить приоритет той или иной целевой функции.

2. Задачи оптимизации. Можно выделить два типа задач оптимизации — безусловные и условные. *Безусловная задача* оптимизации состоит в отыскании максимума или минимума действительной функции (6.1) от n действительных переменных и определении соответствующих значений аргументов на некотором множестве σ n-мерного пространства. Обычно рассматриваются задачи минимизации; к ним легко сводятся и задачи на поиск максимума путем замены знака целевой функции на противоположный.

Условные задачи оптимизации, или *задачи с ограничениями*, — это такие, при формулировке которых задаются некоторые условия (ограничения) на множестве σ. Эти ограничения задаются совокупностью некоторых функций, удовлетворяющих уравнениям или неравенствам.

Ограничения-равенства выражают зависимость между проектными параметрами, которая должна учитываться при нахождении решения. Эти ограничения отражают законы природы, наличие ресурсов, финансовые требования и т. п.

В результате ограничений область проектирования σ, определяемая всеми n проектными параметрами, может быть существенно уменьшена в соответствии с физической сущностью задачи. Число M ограничений-равенств может быть произвольным. Их можно записать в виде

$$\begin{aligned} g_1(x_1, x_2, \ldots, x_n) &= 0, \\ g_2(x_1, x_2, \ldots, x_n) &= 0, \\ \ldots\ldots\ldots\ldots\ldots & \\ g_m(x_1, x_2, \ldots, x_n) &= 0. \end{aligned} \tag{6.2}$$

В ряде случаев из этих соотношений можно выразить одни проектные параметры через другие. Это позволяет исключить некоторые параметры из процесса оптимизации, что приводит к уменьшению размерности задачи и облегчает ее решение. Аналогично могут вводиться также *ограничения-неравенства*, имеющие вид

$$\begin{aligned} a_1 \leqslant \varphi_1(x_1, x_2, \ldots, x_n) \leqslant b_1, \\ a_2 \leqslant \varphi_2(x_1, x_2, \ldots, x_n) \leqslant b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_k \leqslant \varphi_k(x_1, x_2, \ldots, x_n) \leqslant b_k. \end{aligned} \tag{6.3}$$

Следует отметить особенность в отыскании решения при наличии ограничений. Оптимальное решение здесь может соответствовать либо локальному экстремуму (максимуму или минимуму) внутри области проектирования, либо значению целевой функции на границе области. Если же

ограничения отсутствуют, то ищется оптимальное решение на всей области проектирования, т. е. глобальный экстремум.

Теория и методы решения задач оптимизации при наличии ограничений составляют предмет исследования одного из важных разделов прикладной математики — *математического программирования*, некоторые элементы которого будут рассмотрены в § 4.

3. Пример постановки задачи. Пусть требуется спроектировать контейнер в форме прямоугольного параллелепипеда объемом $V = 1$ м^3, причем желательно израсходовать на его изготовление как можно меньше материала.

При постоянной толщине стенок последнее условие означает, что площадь полной поверхности контейнера должна быть минимальной. Если обозначить через x_1, x_2, x_3 длины ребер контейнера, то задача сведется к минимизации функции

$$S = 2(x_1x_2 + x_2x_3 + x_1x_3).$$

Эта функция в данном случае является целевой, а условие $V = 1$ — ограничением-равенством, которое позволяет исключить один параметр:

$$V = x_1x_2x_3 = 1, \quad x_3 = \frac{1}{x_1x_2},$$
$$S = 2\left(x_1x_2 + \frac{1}{x_1} + \frac{1}{x_2}\right). \tag{6.4}$$

Задача свелась к минимизации функции двух переменных. В результате решения задачи будут найдены значения проектных параметров x_1, x_2, а затем и x_3. В приведенном примере фактически получилась задача безусловной оптимизации для целевой функции (6.4), поскольку ограничение-равенство было использовано для исключения параметра x_3.

Вместе с тем можно усложнить рассматриваемую задачу и поставить дополнительные условия. Например, потребуем, чтобы данный контейнер имел длину не менее 2 м. Это условие запишется в виде ограничения-неравенства на один из параметров, например

$$x_1 \geqslant 2. \tag{6.5}$$

Таким образом, мы получили следующую условную задачу оптимизации: минимизируя функцию (6.4) и учитывая ограничение-неравенство (6.5), найти оптимальные значения параметров плана x_1, x_2 ($x_1 \geqslant 0$, $x_2 \geqslant 0$).

§ 2. Одномерная оптимизация

1. Задачи на экстремум. *Одномерная задача оптимизации* в общем случае формулируется следующим образом. Найти наименьшее (или наибольшее) значение целевой функции $y = f(x)$, заданной на множестве σ,

и определить значение проектного параметра $x \in \sigma$, при котором целевая функция принимает экстремальное значение. Существование решения поставленной задачи вытекает из следующей теоремы.

Теорема Вейерштрасса. *Всякая функция $f(x)$, непрерывная на отрезке $[a, b]$, принимает на этом отрезке наименьшее и наибольшее значения, т. е. на отрезке $[a, b]$ существуют такие точки x_1 и x_2, что для любого $x \in [a, b]$ имеют место неравенства*

$$f(x_1) \leqslant f(x) \leqslant f(x_2).$$

Эта теорема не доказывает единственности решения. Не исключена возможность достижения равных экстремальных значений сразу в нескольких точках данного отрезка. В частности, такая ситуация имеет место для периодической функции, рассматриваемой на отрезке, содержащем несколько периодов.

Будем рассматривать методы оптимизации для разных классов целевых функций. Простейшим из них является случай дифференцируемой функции $f(x)$ на отрезке $[a, b]$, причем функция задана в виде аналитической зависимости $y = f(x)$, и может быть найдено явное выражение для ее производной $f'(x)$. Нахождение экстремумов таких функций можно проводить известными из курса высшей математики методами дифференциального исчисления. Напомним вкратце этот путь.

Функция $f(x)$ может достигать своего наименьшего и наибольшего значений либо в граничных точках отрезка $[a, b]$, либо в точках минимума и максимума. Последние точки обязательно должны быть критическими, т. е. производная $f'(x)$ в этих точках обращается в нуль, — это необходимое условие экстремума. Следовательно, для определения наименьшего или наибольшего значений функции $f(x)$ на отрезке $[a, b]$ нужно вычислить ее значения во всех критических точках данного отрезка и в его граничных точках и сравнить полученные значения; наименьшее или наибольшее из них и будет искомым значением.

Пример. Найти наименьшее и наибольшее значения функции $f(x) = x^3/3 - x^2$ на отрезке $[1, 3]$.

Решение. Вычислим производную этой функции:

$$f'(x) = x^2 - 2x.$$

Приравнивая ее нулю, найдем критические точки:

$$x^2 - 2x = 0, \quad x_1 = 0, \quad x_2 = 2.$$

Точка $x = 0$ лежит вне рассматриваемого отрезка, поэтому для анализа оставляем три точки: $a = 1$, $x_2 = 2$, $b = 3$. Вычисляем значения функции в этих точках:

$$f(1) = -2/3, \quad f(2) = -4/3, \quad f(3) = 0.$$

Сравнивая полученные величины, находим, что наименьшего значения функция $f(x)$ достигает в точке $x = 2$, наибольшего — в точке $x = 3$, т. е.

$$f_{\min} = f(2) = -4/3, \quad f_{\max} = f(3) = 0.$$

В рассмотренном примере уравнение $f'(x) = 0$ для отыскания критических точек удалось решить непосредственно. Для более сложных видов производной функции $f'(x)$ необходимо использовать численные методы решения нелинейных уравнений. Например, применение метода Ньютона будет рассмотрено в п. 4.

Как уже отмечалось, используемый здесь метод, основанный на вычислении производной целевой функции, требует ее аналитического представления. В других случаях, когда целевая функция задана в табличном виде или может быть вычислена при некоторых дискретных значениях аргумента, используются различные *методы поиска*. Они основаны на вычислении целевой функции в отдельных точках и выборе среди них наибольшего или наименьшего значений. Существует ряд алгоритмов решения данной задачи. Рассмотрим некоторые из них.

2. Методы поиска. Численные методы поиска экстремальных значений функции рассмотрим на примере нахождения минимума функции $f(x)$ на отрезке $[a, b]$. Будем предполагать, что целевая функция *унимодальна*, т. е. на данном отрезке она имеет только один минимум. Отметим, что в инженерной практике обычно встречаются именно такие целевые функции.

Погрешность приближенного решения задачи определяется разностью между оптимальным значением x проектного параметра и приближением к нему x_*. Потребуем, чтобы эта погрешность была по модулю меньше заданного допустимого значения ε:

$$|x - x_*| < \varepsilon. \tag{6.6}$$

Процесс решения задачи методом поиска состоит в последовательном сужении интервала изменения проектного параметра, называемого *интервалом неопределенности*. В начале процесса оптимизации его длина равна $b - a$, а к концу она должна стать меньше ε, т. е. оптимальное значение проектного параметра должно находиться в интервале неопределенности — отрезке $[x_n, x_{n+1}]$, причем $x_{n+1} - x_n < \varepsilon$. Тогда для выполнения (6.6) в качестве приближения к оптимальному значению можно принять любое $x_* \in [x_n, x_{n+1}]$, например, $x_* = x_n$ или $x_* = x_{n+1}$, или $x_* = (x_n + x_{n+1})/2$. В последнем случае для выполнения (6.6) достаточно выполнения неравенства $x_{n+1} - x_n < 2\varepsilon$.

Наиболее простым способом сужения интервала неопределенности является деление его на некоторое число равных частей с последующим вычислением значений целевой функции в точках разбиения. Пусть n — число элементарных отрезков, $h = (b - a)/n$ — шаг разбиения. Вычислим значения целевой функции $y_k = f(x_k)$ в узлах $x_k = a + kh$ $(k = 0, 1, \ldots, n)$. Сравнивая полученные значения $f(x_k)$, найдем среди них наименьшее $y_i = f(x_i)$.

Число $m_n = y_i$ можно приближенно принять за наименьшее значение целевой функции $f(x)$ на отрезке $[a, b]$. Очевидно, что близость m_n к минимуму m зависит от числа точек, и для непрерывной функции $f(x)$

$$\lim_{n \to \infty} m_n = m,$$

т. е. с увеличением числа точек разбиения погрешность в определении минимума стремится к нулю.

В данном методе, который можно назвать *методом перебора,* основная трудность состоит в выборе n и оценке погрешности. Можно, например, провести оптимизацию с разными шагами и исследовать сходимость такого итерационного процесса. Но это трудоемкий путь.

Более экономичным способом уточнения оптимального параметра является использование свойства унимодальности целевой функции, которое позволяет построить процесс сужения интервала неопределенности. Пусть, как и ранее, среди всех значений унимодальной функции $y = f(x)$, вычисленных в узлах x_k $(k = 0, 1, \ldots, n)$, наименьшим оказалось y_i. Это означает, что оптимальное значение проектного параметра находится на отрезке $[x_{i-1}, x_{i+1}]$ (рис. 6.1), т. е. интервал неопределенности сузился до длины двух шагов. Если размер интервала недостаточен для удовлетворения заданной погрешности, т. е. $x_{i+1} - x_{i-1} \geqslant \varepsilon$, то его снова можно уменьшить путем нового разбиения. Получится интервал, равный двум длинам нового шага разбиения, и т. д. Процесс оптимизации продолжается до достижения заданного размера интервала неопределенности. В описанном методе общего поиска можно с помощью некоторой изобретательности, а также разумного выбора шага разбиения добиться эффективного поиска.

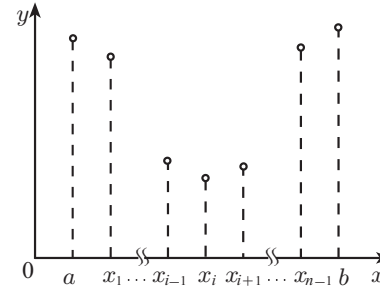


Рис. 6.1

Например, пусть начальная длина интервала неопределенности равна $b - a = 1$. Нужно добиться его уменьшения в 100 раз. Этого легко достичь разбиением интервала на 200 частей. Вычислив значения целевой функции $f(x_k)$ $(k = 0, 1, \ldots, 200)$, найдем ее минимальное значение $f(x_i)$. Тогда искомым интервалом неопределенности будет отрезок $[x_{i-1}, x_{i+1}]$.

Однако можно поступить и иначе. Сначала разобьем отрезок $[a, b]$ на 20 частей и найдем интервал неопределенности длиной 0.1, при этом мы вычислим значения целевой функции в точках $x_k = a + 0.05k$ $(k = 0, 1, \ldots, 20)$. Теперь отрезок $[x_{i-1}, x_{i+1}]$ снова разобьем на 20 частей; получим искомый интервал длиной 0.01, причем значения целевой функции вычисляем в точках $x_k = x_{i-1} + 0.005k$ $(k = 1, 2, \ldots, 19)$ (в точках x_{i-1} и x_{i+1} значения $f(x)$ уже найдены). Таким образом, во втором случае в процессе оптимизации произведено 40 вычислений значений целевой функции против 201 в первом случае, т. е. способ разбиения позволяет получить существенную экономию вычислений.

Существует ряд специальных методов поиска оптимальных решений с разными способами выбора узлов и сужения интервала неопределенности: метод деления отрезка пополам, метод золотого сечения и др. Рассмотрим один из них.

3. Метод золотого сечения. При построении процесса оптимизации стараются сократить объем вычислений и время поиска. Этого достигают обычно путем сокращения количества вычислений (или измерений — при проведении эксперимента) значений целевой функции $f(x)$. Одним из наиболее эффективных методов, в которых при ограниченном количестве вычислений $f(x)$ достигается наилучшая точность, является *метод золотого сечения*. Он состоит в построении последовательности отрезков $[a_0, b_0], [a_1, b_1], \ldots$, стягивающихся к точке минимума функции $f(x)$. На каждом шаге, за исключением первого, вычисление значения функции $f(x)$ проводится лишь в одной точке. Эта точка, называемая *золотым сечением*, выбирается специальным образом.

Поясним сначала идею метода геометрически, а затем выведем необходимые соотношения. На первом шаге процесса оптимизации внутри отрезка $[a_0, b_0]$ (рис. 6.2, *а*) выбираем некоторые внутренние точки x_1 и x_2 и вычисляем значения целевой функции $f(x_1)$ и $f(x_2)$. Поскольку в данном случае $f(x_1) < f(x_2)$, очевидно, что минимум расположен на одном из прилегающих к x_1 отрезков: $[a_0, x_1]$ или $[x_1, x_2]$. Поэтому отрезок $[x_2, b_0]$ можно отбросить, сузив тем самым первоначальный интервал неопределенности.

Второй шаг проводим на отрезке $[a_1, b_1]$ (рис. 6.2, *б*), где $a_1 = a_0$, $b_1 = x_2$. Нужно снова выбрать две внутренние точки, но одна из них (x_1) осталась из предыдущего шага, поэтому достаточно выбрать лишь одну точку x_3, вычислить значение $f(x_3)$ и провести сравнение. Поскольку здесь

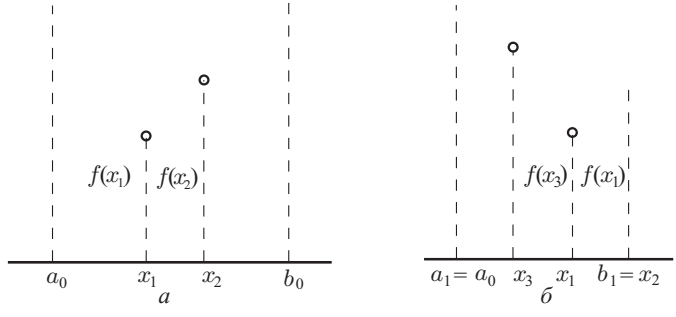


Рис. 6.2

$f(x_3) < f(x_1)$, ясно, что минимум находится на отрезке $[x_3, b_1]$. Обозначим этот отрезок $[a_2, b_2]$, снова выберем одну внутреннюю точку и повторим процедуру сужения интервала неопределенности. Процесс оптимизации повторяется до тех пор, пока длина очередного отрезка $[a_k, b_k]$ не станет меньше заданной величины ε.

Теперь рассмотрим способ размещения внутренних точек на каждом отрезке $[a_k, b_k]$. Пусть длина интервала неопределенности равна l, а точка деления разбивает его на части l_1, l_2: $l_1 > l_2$, $l = l_1 + l_2$. *Золотое сечение* интервала неопределенности выбирается так, чтобы отношение длины

большего отрезка к длине всего интервала равнялось отношению длины меньшего отрезка к длине большего отрезка:

$$\frac{l_1}{l} = \frac{l_2}{l_1}. \tag{6.7}$$

Из этого соотношения можно найти точку деления, вычислив отношения

$$\alpha = \frac{l_1}{l}, \quad \beta = \frac{l_2}{l}.$$

Преобразуем выражение (6.7) и найдем значения α, β:

$$l_1^2 = l_2 l, \quad l_1^2 = l(l - l_1), \quad l_1^2 + l_1 l - l^2 = 0,$$

$$\left(\frac{l_1}{l}\right)^2 + \frac{l_1}{l} - 1 = 0, \quad \alpha^2 + \alpha - 1 = 0, \quad \alpha = \frac{-1 \pm \sqrt{5}}{2}.$$

Поскольку нас интересует только положительное решение, то

$$\alpha = \frac{-1 + \sqrt{5}}{2} \approx 0.618, \quad \beta = 1 - \alpha = \frac{3 - \sqrt{5}}{2} \approx 0.382.$$

Очевидно, что интервал неопределенности можно разделить в соотношении золотого сечения двояко: в пропорциях $l_2 : l_1$ и $l_1 : l_2$. На рис. 6.2, *а* точки деления x_1 и x_2 выбираются с учетом этих пропорций. В данном случае имеем

$$\frac{x_1 - a_0}{b_0 - a_0} = \frac{l_2}{l} = \beta, \quad x_1 - a_0 = \beta(b_0 - a_0), \quad x_1 = (1 - \beta)a_0 + \beta b_0,$$

$$x_1 = \alpha a_0 + \beta b_0. \tag{6.8}$$

Аналогично,

$$x_2 = \beta a_0 + \alpha b_0. \tag{6.9}$$

Начальная длина интервала неопределенности составляет $d_0 = b_0 - a_0$. После первого шага оптимизации получается новый интервал неопределенности — отрезок $[a_1, b_1]$ (см. рис. 6.2, *б*). Его длина с учетом (6.9) равна

$$d_1 = b_1 - a_1 = x_2 - a_0 = \beta a_0 + \alpha b_0 - a_0 = \alpha(b_0 - a_0) = \alpha d_0 \approx 0.618\, d_0.$$

На втором шаге отрезок $[a_1, b_1]$ также делится с соотношении золотого сечения. При этом одной из точек деления будет точка x_1. Покажем это:

$$\frac{x_1 - a_0}{x_2 - a_0} = \frac{\beta(b_0 - a_0)}{\alpha(b_0 - a_0)} = \frac{\beta}{\alpha} = \frac{1 - \alpha}{\alpha} = \alpha.$$

Последнее равенство следует из соотношения $\alpha^2 + \alpha - 1 = 0$.

Вторая точка деления x_3 выбирается так же, как выбирается точка x_1 при делении отрезка $[a_0, b_0]$, т. е. аналогично (6.8): $x_3 = \alpha a_1 + \beta b_1$. И снова интервал неопределенности уменьшается до размера

$$d_2 = b_2 - a_2 = b_1 - x_3 = b_1 - \alpha a_1 - \beta b_1 = \alpha(b_1 - a_1) = \alpha d_1 = \alpha^2 d_0.$$

По аналогии с соотношениями (6.8), (6.9) можно записать координаты точек деления y и z отрезка $[a_{k-1}, b_{k-1}]$ на k-м шаге оптимизации ($y < z$):

$$y = \alpha a_{k-1} + \beta b_{k-1},$$
$$z = \beta a_{k-1} + \alpha b_{k-1}.$$

Вычислению, естественно, подлежит только одна из координат y, z; другая координата берется с предыдущего шага. При этом длина интервала неопределенности равна

$$d_k = b_k - a_k = \alpha^k d_0 \approx 0.618^k\, d_0. \tag{6.10}$$

Как и в общем случае метода поиска, процесс оптимизации заканчивается при выполнении условия $d_k < \varepsilon$. Тогда проектный параметр оптимизации $x \in [a_k, b_k]$. В качестве приближения к оптимальному значению можно принять $x_* = a_k$ или $x_* = b_k$, или $x_* = (a_k + b_k)/2$. В последнем случае для достижения требуемой точности (для выполнения неравенства (6.6)) достаточно, чтобы

$$d_k < 2\varepsilon. \tag{6.11}$$

На рис. 6.3 представлен алгоритм процесса одномерной оптимизации методом золотого сечения. Здесь y, z — точки деления отрезка $[a, b]$, причем $y < z$. Переменная q используется для выхода из цикла при выполнении неравенства (6.11), т. е. после достижения требуемой точности. В результате выполнения алгоритма выдается оптимальное значение проектного параметра x, в качестве которого принимается середина последнего интервала неопределенности.

Пример. Для оценки сопротивления дороги движению автомобиля при скорости v км/ч можно использовать эмпирическую формулу $f(v) = 24 - \dfrac{2}{3}\,v + \dfrac{1}{30}\,v^2$ (для шоссе). Определить скорость, при которой сопротивление будет минимальным.

Решение. Это простейшая задача одномерной оптимизации. Здесь сопротивление $f(v)$ — целевая функция, скорость v — проектный параметр. Данную задачу легко решить путем нахождения минимума с помощью вычисления производной, поскольку $f(v)$ — функция дифференцируемая. Действительно,

$$f'(v) = -\frac{2}{3} + \frac{2}{30}\,v = 0, \quad v = 10 \text{ км/ч}.$$

Проиллюстрируем на этой простейшей задаче метод золотого сечения. Первоначально границы интервала неопределенности примем равными $a = 5$, $b = 20$. Результаты вычислений представим в виде (табл. 6.1). Здесь обозначения аналогичны используемым в структурограмме (см. рис. 6.3). Расчеты проводятся в соответствии со структурограммой с погрешностью $\varepsilon = 1$ км/ч.

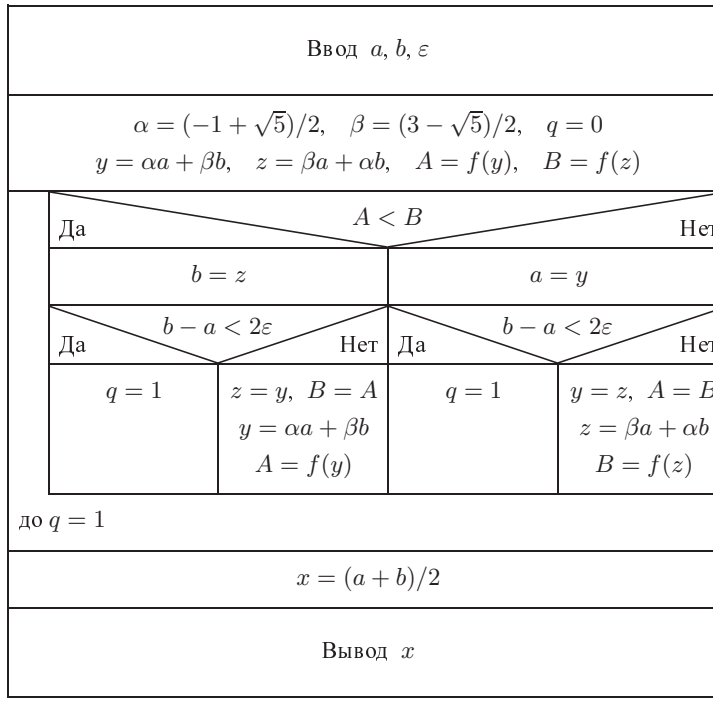


Рис. 6.3. Метод золотого сечения

Приведем решение для первого этапа:

$$y = \alpha \cdot 5 + \beta \cdot 20 \approx 0.618 \cdot 5 + 0.382 \cdot 20 \approx 10.7,$$

$$z = \beta \cdot 5 + \alpha \cdot 20 \approx 0.382 \cdot 5 + 0.618 \cdot 20 \approx 14.3,$$

$$A \approx 24 - \frac{2}{3} \cdot 10.7 + \frac{1}{30} \cdot 10.7^2 \approx 20.7,$$

$$B \approx 24 - \frac{2}{3} \cdot 14.3 + \frac{1}{30} \cdot 14.3^2 \approx 21.3,$$

$$A < B.$$

При данной невысокой точности вычислений достаточно шести шагов оптимизации (согласно алгоритму, приведенному на рис. 6.3, последний шаг выполняется не полностью). В этом случае приближенное искомое значение скорости равно

$$v_* = \frac{9.4 + 10.7}{2} = 10.05 \text{ км/ч}.$$

Замечание. Согласно табл. 6.1 заданная точность ε будет достигнута уже на третьем шаге:

$$v_* = \frac{8.6 + 12.1}{2} = 10.35 \text{ км/ч}, \quad |10 - 10.35| < \varepsilon = 1.$$

Однако определить достижение заданной точности на третьем шаге можно только тогда, когда известно точное решение $v = 10$ км/ч. Но в этом

Таблица 6.1

Шаг	a	y	z	b	A	B	$b-a$
1	5	10.7	14.3	20	20.7	21.3	15
2	5	8.6	10.7	14.3	20.73	20.68	9.3
3	8.6	10.7	12.1	14.3	20.68	20.81	5.7
4	8.6	9.9	10.7	12.1	20.66	20.68	3.5
5	8.6	9.4	9.9	10.7	20.68	20.66	2.1
6	9.4			10.7			1.3

случае вообще нет смысла в проведении численного расчета. Поэтому в реальном расчете нужно выполнить большее число шагов до выполнения условия (6.11).

Метод золотого сечения (как и, например, метод решения нелинейных уравнений делением отрезка пополам, см. гл. 5, § 1, п. 2) относится к тем немногим численным методам, для которых можно гарантировать, что требуемая точность достигнута. Используя соотношения (6.10) и (6.11), можно найти число итераций N, требуемое для достижения точности ε (ср. с (5.7)):

$$N = E\left(\log_\alpha \frac{2\varepsilon}{b-a}\right) + 2. \tag{6.12}$$

Для рассмотренного выше примера (6.12) дает $N = 6$.

4. Метод Ньютона. Как было отмечено в п. 1, задача одномерной оптимизации дифференцируемой функции $f(x)$ сводится к нахождению критических точек этой функции, определяемых уравнением

$$f'(x) = 0. \tag{6.13}$$

Когда уравнение (6.13) нельзя решить аналитически, для его решения можно применить численные методы, например метод Ньютона (см. гл. 5, § 1, п. 4). В этом случае говорят о *методе Ньютона решения задачи оптимизации*.

Пусть $x = c$ — решение уравнения (6.13), а c_0 — некоторое начальное приближение к c. Применим для решения (6.13) метод Ньютона решения уравнения $F(x) = 0$, которое эквивалентно уравнению (6.13) при $F(x) = f'(x)$. Для этого в формулу для k-го приближения метода Ньютона (5.11)

подставим вместо $F(x)$ производную $f'(x)$ и получим тем самым формулу для k-го приближения к решению уравнения (6.13):

$$c_k = c_{k-1} - f'(c_{k-1})/f''(c_{k-1}), \quad k = 1, 2, \ldots \tag{6.14}$$

Для использования этой формулы необходимо, чтобы $f''(c_{k-1}) \neq 0$. В качестве критерия окончания итерационного процесса можно применять условия близости двух последовательных приближений

$$|c_k - c_{k-1}| < \varepsilon$$

или близости значений целевой функции на этих приближениях

$$|f(c_k) - f(c_{k-1})| < \varepsilon.$$

Достаточное условие сходимости метода Ньютона (6.14) можно получить, опираясь на теорему из п. 4 § 1 гл. 5. А именно, справедлива следующая теорема.

Теорема. *Пусть $x = c$ — корень уравнения* (6.13), *т. е. $f'(c) = 0$, а $f''(c) \neq 0$ и $f'''(x)$ непрерывна. Тогда существует окрестность D корня c ($c \in D$) такая, что если начальное приближение c_0 принадлежит этой окрестности, то для метода Ньютона* (6.14) *последовательность значений $\{c_k\}$ сходится к c при $k \to \infty$.*

Заметим, что точка $x = c$ может являться как точкой минимума, так и точкой максимума, а может (при $f''(c) = 0$) вообще не являться точкой экстремума. Если функция $f(x)$ имеет как минимумы, так и максимумы, то c_k может сходиться и к точкам минимума, и к точкам максимума в зависимости от того, из окрестности какой критической точки взято начальное приближение. При этом, в отличие от других методов оптимизации, формула для поиска максимума функции совпадает с формулой для поиска минимума.

Формулу метода Ньютона решения задачи оптимизации можно получить и из других соображений. Разложим функцию $f(x)$ в ряд Тейлора в окрестности точки c_{k-1}, ограничившись линейными и квадратичными членами относительно приращения $x - c_{k-1}$:

$$f(x) \approx \varphi(x) = f(c_{k-1}) + f'(c_{k-1})(x - c_{k-1}) + \frac{1}{2} f''(c_{k-1})(x - c_{k-1})^2. \tag{6.15}$$

В качестве следующего приближения c_k к оптимальному значению проектного параметра x возьмем точку экстремума функции $\varphi(x)$. Имеем

$$\varphi'(c_k) = f'(c_{k-1}) + f''(c_{k-1})(x - c_{k-1}) = 0,$$

$$c_k = c_{k-1} - \frac{f'(c_{k-1})}{f''(c_{k-1})},$$

что совпадает с (6.14). Разложение (6.15) в окрестности точки c_0 иллюстрирует рис. 6.4, на котором график функции $f(x)$ заменяется параболой — графиком функции $\varphi(x)$.

Относительно сходимости метода Ньютона решения задачи оптимизации можно сделать замечания, аналогичные сделанным в гл. 5. Метод Ньютона обладает более быстрой сходимостью по сравнению с методами, которые не используют дифференциальные свойства функции (например, с методом золотого сечения). Однако сходимость метода Ньютона не гарантирована, при неудачном выборе начального приближения он может расходиться.

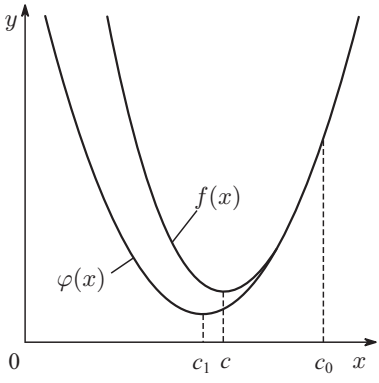


Рис. 6.4. Метод Ньютона

Пример. Решим методом Ньютона задачу оптимизации из п. 3.

Решение. Заметим, что функция $f(v) = 24 - \frac{2}{3}v + \frac{1}{30}v^2$ является квадратичной, т. е. ее разложение вида (6.15) представляет собой точное равенство: $f(v) = \varphi(v)$. Поэтому первая же итерация по методу Ньютона $v_* = c_1$ при любом начальном приближении c_0 даст точное решение задачи. Второе приближение совпадет с первым: $c_2 = c_1$, поскольку $f'(c_1) = 0$. Таким образом, после двух итераций будет выполнено условие завершения итерационного процесса.

§ 3. Многомерные задачи оптимизации

1. Минимум функции нескольких переменных. В § 2 мы рассмотрели одномерные задачи оптимизации, в которых целевая функция зависит лишь от одного аргумента. Однако в большинстве реальных задач оптимизации, представляющих практический интерес, целевая функция зависит от многих проектных параметров. В частности, рассмотренная выше задача об определении сопротивления дороги движению автомобиля на самом деле является многомерной, поскольку здесь наряду со скоростью имеются и другие проектные параметры (качество покрытия, уклон, температура и др.).

Минимум дифференцируемой функции многих переменных $u = f(x_1, x_2, \ldots, x_n)$ можно найти, исследуя ее значения в критических точках, которые определяются из решения системы дифференциальных уравнений

$$\frac{\partial f}{\partial x_1} = 0, \quad \frac{\partial f}{\partial x_2} = 0, \quad \ldots, \quad \frac{\partial f}{\partial x_n} = 0. \tag{6.16}$$

Пример. В § 1 (п. 3) была рассмотрена задача об определении оптимальных размеров контейнера объем которого равен 1 м^3. Задача свелась к

минимизации полной поверхности контейнера, которая в данном случае является целевой функцией

$$S = 2\left(x_1 x_2 + \frac{1}{x_1} + \frac{1}{x_2}\right).$$

Р е ш е н и е . В соответствии с (6.16) получим систему

$$\frac{\partial S}{\partial x_1} = 2\left(x_2 - \frac{1}{x_1^2}\right) = 0,$$
$$\frac{\partial S}{\partial x_2} = 2\left(x_1 - \frac{1}{x_2^2}\right) = 0.$$

Отсюда находим $x_1 = x_2 = 1$ м, $x_3 = 1/(x_1 x_2) = 1$ м. Таким образом, оптимальной формой контейнера в данном случае является куб, длина ребра которого равна 1 м.

Рассмотренный метод можно использовать лишь для дифференцируемой целевой функции. Но и в этом случае могут возникнуть серьезные трудности при решении системы нелинейных уравнений (6.16).

Во многих случаях никакой формулы для целевой функции нет, а имеется лишь возможность определения ее значений в произвольных точках рассматриваемой области с помощью некоторого вычислительного алгоритма или путем физических измерений. Задача состоит в приближенном определении наименьшего значения функции во всей области при известных ее значениях в отдельных точках.

Для решения подобной задачи в области проектирования G, в которой ищется минимум целевой функции $u = f(x_1, x_2, \ldots, x_n)$, можно ввести дискретное множество точек (узлов) путем разбиения интервалов изменения параметров x_1, x_2, $\ldots$, x_n на части с шагами h_1, h_2, $\ldots$, h_n. В полученных узлах можно вычислить значения целевой функции и среди этих значений найти наименьшее.

Такой метод аналогичен методу перебора для функции одной переменной (см. § 2, п. 2). Однако в многомерных задачах оптимизации, где число проектных параметров достигает пяти и более, этот метод потребовал бы слишком большого объема вычислений.

Оценим, например, объем вычислений с помощью перебора при решении задачи оптимизации функции пяти неизвестных. Пусть вычисление ее значения в одной точке требует 100 арифметических операций (на практике это число может достигать нескольких тысяч и больше). Область проектирования разделим на 100 частей в каждом из пяти направлений, т. е. число расчетных точек равно $101^5 \approx 10^{10}$. Число арифметических операций тогда равно 10^{12}, и для решения этой задачи на компьютере с быстродействием 10 млн. оп./с потребуется 10^5 с (более суток) машинного времени.

Проведенная оценка показывает, что такие методы общего поиска с использованием сплошного перебора для решения многомерных задач оптимизации не годятся. Необходимы специальные численные методы, основанные на целенаправленном поиске. Рассмотрим некоторые из них.

2. Метод покоординатного спуска. Пусть требуется найти наименьшее значение целевой функции $u = f(x_1, x_2, \ldots, x_n)$. В качестве начального приближения выберем в n–мерном пространстве некоторую точку M_0 с координатами $x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$. Зафиксируем все координаты функции u, кроме первой. Тогда $v(x_1) = f\big(x_1, x_2^{(0)}, \ldots, x_n^{(0)}\big)$ — функция одной переменной x_1. Первый шаг процесса оптимизации состоит в спуске по координате x_1 в направлении убывания функции v от точки M_0 до некоторой точки $M_1\big(x_1^{(1)}, x_2^{(0)}, \ldots, x_n^{(0)}\big)$. Если функция f дифференцируемая, то значение $x_1^{(1)}$ может быть найдено как

$$x_1^{(1)} = x_1^{(0)} - \alpha_1^{(1)} \frac{\partial f}{\partial x_1}(M_0). \tag{6.17}$$

Здесь $\alpha_1^{(1)} > 0$ — некоторый шаг. Соотношение (6.17) определяет движение в сторону уменьшения значений функции v (если только шаг $\alpha_1^{(1)}$ не слишком велик, в противном случае его нужно уменьшить). Действительно, пусть $\partial f/\partial x_1(M_0) > 0$. Тогда с ростом x_1 функция v возрастает, а (6.17) определяет движение в сторону уменьшения x_1.

Значение $x_1^{(1)}$ можно найти и иначе. А именно, можно решить одномерную задачу оптимизации для функции $v(x_1)$. Тогда функция v в точке M_1 примет наименьшее значение, т. е. функция u примет в этой точке наименьшее значение по координате x_1 при фиксированных остальных координатах.

Зафиксируем теперь все координаты, кроме x_2, и рассмотрим функцию этой переменной $w(x_2) = f\big(x_1^{(1)}, x_2, x_3^{(0)}, \ldots, x_n^{(0)}\big)$. Снова осуществляем спуск, теперь по координате x_2, в сторону убывания функции w от точки M_1 до точки $M_2\big(x_1^{(1)}, x_2^{(1)}, x_3^{(0)}, \ldots, x_n^{(0)}\big)$. Значение $x_2^{(1)}$ можно найти либо как

$$x_2^{(1)} = x_2^{(0)} - \alpha_2^{(1)} \frac{\partial f}{\partial x_2}(M_1),$$

либо, решив задачу одномерной оптимизации для функции $w(x_2)$.

Аналогично проводится спуск по координатам x_3, x_4, …, x_n, а затем процедура снова повторяется от x_1 до x_n и т. д. В результате этого процесса получается последовательность точек M_0, M_1, …, в которых значения целевой функции составляют монотонно убывающую последовательность $f(M_0) \geqslant f(M_1) \geqslant \ldots$

В качестве критерия завершения итерационного процесса можно использовать условия близости значений проектных параметров или целевой функции на двух последовательных итерациях. Однако под итерацией здесь следует понимать всю процедуру спуска по координатам от x_1 до x_n. Таким образом, близость проектных параметров можно трактовать как выполнение условий (4.21) – (4.23), а близость значений целевой функции

как выполнение условия

$$\left| f\left(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\right) - f\left(x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_n^{(k-1)}\right) \right| < \varepsilon. \qquad (6.18)$$

Метод покоординатного спуска сводит задачу о нахождении наименьшего значения функции многих переменных к многократному спуску в сторону уменьшения значений функции по каждому проектному параметру. Данный метод легко проиллюстрировать геометрически для случая функции двух переменных $z = f(x, y)$, описывающей некоторую поверхность в трехмерном пространстве. На рис 6.5 нанесены линии уровня этой поверхности. Процесс оптимизации в этом случае проходит следующим образом. Точка $M_0(x_0, y_0)$ описывает начальное приближение. Проводя спуск по координате x, попадем в точку $M_1(x_1, y_0)$. Далее, двигаясь параллельно оси ординат, придем в точку $M_2(x_1, y_1)$ и т. д.

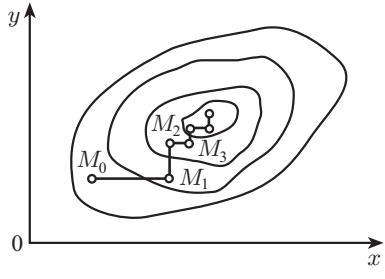


Рис. 6.5. Спуск по координатам

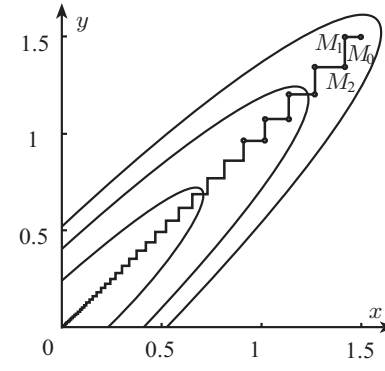


Рис. 6.6. Овраг на поверхности

Важным здесь является вопрос о сходимости рассматриваемого процесса оптимизации. Другими словами, будет ли последовательность значений целевой функции $f(M_0), f(M_1), \ldots$ сходиться к наименьшему ее значению в данной области и если будет, то как быстро? Это зависит от вида самой функции и выбора начального приближения.

Для функции двух переменных очевидно, что метод может оказаться неприменимым при наличии изломов в линиях уровня. Для гладких функций при удачно выбранном начальном приближении (в некоторой окрестности минимума) процесс сходится к минимуму. Здесь однако применение метода затруднено в случае так называемых *оврагов* на поверхности.

Овраг представляет собой впадину, линии уровня которой имеют форму овалов с различающимися во много раз длинами осей. Пример оврага показан на рис. 6.6 (рисунок ограничен первой координатной четвертью),

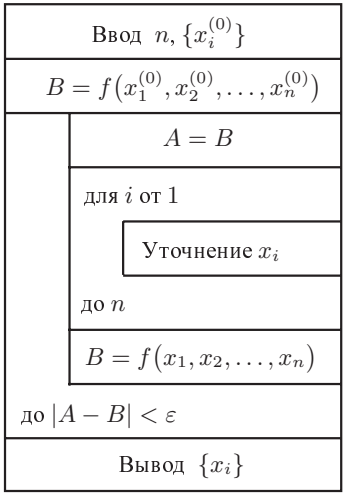


Рис. 6.7. Структурограмма метода покоординатного спуска

изображены линии уровня функции, которую поворотом системы координат можно привести к виду $z = x^2 + 36y^2$. Точкой минимума этой функции является начало координат. Наличие оврага приводит к тому, что процесс спуска к минимуму очень длительный, метод сходится медленно. Так при начальном приближении $x = y = 1.5$ после 50 итераций получается приближение $x \approx y \approx 0.006$.

Поскольку поверхности типа «оврага» встречаются в инженерной практике, то при использовании метода покоординатного спуска следует убедиться, что решаемая задача не имеет этого недостатка.

К достоинствам метода покоординатного спуска следует отнести возможность использования простых алгоритмов одномерной оптимизации. Структурограмма метода покоординатного спуска представлена на рис. 6.7. Предполагается, что итерации завершаются при выполнении условия (6.18).

3. Метод градиентного спуска. В природе мы нередко наблюдаем явления, сходные с решением задачи на нахождение минимума. К ним относится, в частности, стекание воды с берега котлована на дно. Упростим ситуацию, считая, что берега котлована «унимодальны», т. е. они гладкие и не содержат локальных углублений или выступов. Тогда вода устремится вниз в направлении наибольшей крутизны берега в каждой точке.

Переходя на математический язык, заключаем, что направление наискорейшего спуска соответствует направлению наибольшего убывания функции. Из курса математики известно, что направление наибольшего возрастания функции двух переменных $u = f(x, y)$ характеризуется ее *градиентом*

$$\operatorname{grad} u = \frac{\partial u}{\partial x}\, e_1 + \frac{\partial u}{\partial y}\, e_2,$$

где e_1, e_2 — единичные векторы (орты) в направлении координатных осей. Следовательно, направление, противоположное градиентному, укажет направление наибольшего убывания функции. Методы, основанные на выборе пути оптимизации с помощью градиента, называются *градиентными*.

Идея метода градиентного спуска состоит в следующем. Выбираем некоторую начальную точку $M_0(\mathbf{x}^{(0)})$, $\mathbf{x}^{(0)} = \{x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\}$, и вычисляем в ней градиент рассматриваемой функции. Делаем шаг в направлении, обратном градиентному:

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha^{(1)} \operatorname{grad} f(M_0).$$

В результате приходим в точку $M_1\left(\mathbf{x}^{(1)}\right)$, значение функции в которой обычно меньше первоначального ($\alpha^{(1)} > 0$). Если это условие не выполнено, т. е. значение функции не изменилось либо даже возросло, то нужно уменьшить шаг $\alpha^{(1)}$. В новой точке процедуру повторяем: вычисляем градиент и снова делаем шаг в обратном к нему направлении:

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha^{(2)} \operatorname{grad} f(M_1).$$

Процесс продолжается до получения наименьшего значения целевой функции. Строго говоря, момент окончания поиска наступит тогда, когда движение из полученной точки с любым шагом приводит к возрастанию значения целевой функции. Если минимум функции достигается внутри рассматриваемой области, то в этой точке градиент равен нулю, что также может служить сигналом об окончании процесса оптимизации. Приближенно момент окончания поиска можно определить аналогично тому, как это делается в других итерационных методах. Например, можно проверить близость значений целевой функции на двух последовательных итерациях:

$$\left| f\left(\mathbf{x}^{(k)}\right) - f\left(\mathbf{x}^{(k-1)}\right) \right| < \varepsilon.$$

Метод градиентного спуска обладает тем же недостатком, что и метод покоординатного спуска: при наличии оврагов на поверхности сходимость метода очень медленная.

В описанном методе требуется вычислять на каждом шаге оптимизации градиент целевой функции $f(\mathbf{x})$:

$$\operatorname{grad} f = \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right\}.$$

Формулы для частных производных можно получить в явном виде лишь в том случае, когда целевая функция задана аналитически. В противном случае эти производные вычисляются с помощью численного дифференцирования:

$$\frac{\partial f}{\partial x_i} \approx \frac{1}{\Delta x_i} \left[f(x_1, \ldots, x_i + \Delta x_i, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n) \right],$$
$$i = 1, 2, \ldots, n.$$

При использовании градиентного спуска в задачах оптимизации основной объем вычислений приходится обычно на вычисление градиента целевой функции в каждой точке траектории спуска. Поэтому целесообразно уменьшить количество таких точек без ущерба для самого решения. Это достигается в некоторых методах, являющихся модификациями градиентного спуска. Одним из них является *метод наискорейшего спуска*. Согласно этому методу, после определения в начальной точке направления, противоположного градиенту целевой функции, решают одномерную задачу оптимизации, минимизируя функцию вдоль этого направления. А именно, минимизируется функция

$$g(\alpha) = f\left(\mathbf{x}^{(0)} - \alpha \operatorname{grad} f(M_0)\right).$$

Для минимизации $g(\alpha)$ можно использовать один из методов одномерной оптимизации. Можно и просто двигаться в направлении, противоположном градиенту, делая при этом не один шаг, а несколько шагов до тех пор, пока целевая функция не перестанет убывать. В найденной новой точке снова определяют направление спуска (с помощью градиента) и ищут новую точку минимума целевой функции и т. д. В этом методе спуск происходит гораздо более крупными шагами, и градиент функции вычисляется в меньшем числе точек.

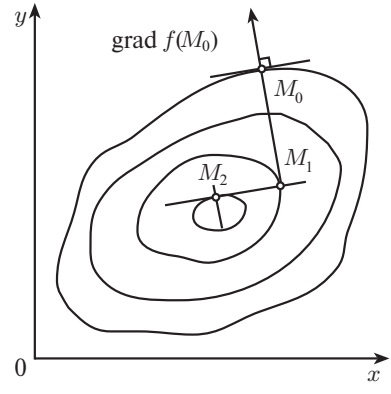


Рис. 6.8. Метод наискорейшего спуска

Заметим, что сведение многомерной задачи оптимизации к последовательности одномерных задач на каждом шаге оптимизации рассмотрено в п. 2 для метода покоординатного спуска. Разница состоит в том, что здесь направление одномерной оптимизации определяется градиентом целевой функции, тогда как покоординатный спуск проводится на каждом шаге вдоль одного из координатных направлений.

Проиллюстрируем метод наискорейшего спуска на рис. 6.8 для случая функции двух переменных $z=f(x,y)$ и отметим некоторые его геометрические особенности.

Во-первых, легко показать, что градиент функции перпендикулярен касательной к линии уровня в данной точке. Следовательно, в градиентных методах спуск происходит по нормали к линии уровня.

Во-вторых, в точке, в которой достигается минимум целевой функции вдоль направления, производная функции по этому направлению обращается в нуль. Но производная функции равна нулю по направлению касательной к линии уровня. Отсюда следует, что градиент целевой функции в новой точке перпендикулярен направлению одномерной оптимизации на предыдущем шаге, т. е. спуск на двух последовательных шагах производится во взаимно перпендикулярных направлениях.

Советуем читателю для лучшего понимания метода наискорейшего спуска составить алгоритм, аналогичный представленному на рис. 6.7 для метода покоординатного спуска.

§ 4. Задачи с ограничениями

1. Метод штрафных функций. Решение задач математического программирования значительно более трудоемко по сравнению с задачами безусловной оптимизации. Ограничения типа равенств или неравенств требуют их учета на каждом шаге оптимизации. Одним из направлений

в методах решения задач математического программирования является сведение их к последовательности задач безусловной минимизации. К этому направлению относится, в частности *метод штрафных функций*.

Сущность метода состоит в следующем. Пусть $f(x_1, x_2, \ldots, x_n)$ — целевая функция, для которой нужно найти минимум m в ограниченной области D, $(x_1, x_2, \ldots, x_n) \in D$. Данную задачу заменяем задачей о безусловной минимизации однопараметрического семейства функций

$$F(\mathbf{x}, \beta) = f(\mathbf{x}) + \frac{1}{\beta}\,\varphi(\mathbf{x}), \quad \mathbf{x} = \{x_1, x_2, \ldots, x_n\}.$$

При этом дополнительную (*штрафную*) функцию $\varphi(\mathbf{x})$ выберем таким образом, чтобы при $\beta \to 0$ решение вспомогательной задачи стремилось к решению исходной или, по крайней мере, чтобы их минимумы совпадали: $\min F(\mathbf{x}, \beta) \to m$ при $\beta \to 0$.

Штрафная функция $\varphi(\mathbf{x})$ должна учитывать ограничения, которые задаются при постановке задачи оптимизации. В частности, если имеются ограничения-неравенства вида $h_j(x_1, x_2, \ldots, x_n) \geqslant 0$ $(j = 1, 2, \ldots, J)$, то в качестве штрафной можно взять функцию, которая:

1) равна нулю во всех точках пространства проектирования, удовлетворяющих заданным ограничениям-неравенствам;

2) стремится к бесконечности в тех точках, в которых эти неравенства не выполняются.

Таким образом, при выполнении ограничений-неравенств функции $f(\mathbf{x})$ и $F(\mathbf{x}, \beta)$ имеют один и тот же минимум. Если хотя бы одно неравенство не выполнится, то вспомогательная целевая функция $F(\mathbf{x}, \beta)$ получает бесконечно большие добавки, и ее значения далеки от минимума функции $f(\mathbf{x})$. Другими словами, при несоблюдении ограничений-неравенств налагается «штраф». Отсюда и термин «метод штрафных функций».

Теперь рассмотрим случай, когда в задаче оптимизации заданы ограничения двух типов — равенства и неравенства:

$$\begin{aligned} &g_i(\mathbf{x}) = 0, \quad i = 1, 2, \ldots, I; \\ &h_j(\mathbf{x}) \geqslant 0, \quad j = 1, 2, \ldots, J; \quad \mathbf{x} = \{x_1, x_2, \ldots, x_n\}. \end{aligned} \tag{6.19}$$

В этом случае в качестве вспомогательной целевой функции, для которой формулируется задача безусловной оптимизации во всем n-мерном пространстве, принимают функцию

$$F(\mathbf{x}, \beta) = f(\mathbf{x}) + \frac{1}{\beta}\left\{ \sum_{i=1}^{I} g_i^2(\mathbf{x}) + \sum_{j=1}^{J} h_j^2(\mathbf{x})\bigl[1 - \operatorname{sign} h_j(\mathbf{x})\bigr] \right\}, \quad \beta > 0. \tag{6.20}$$

Здесь взята такая штрафная функция, что при выполнении условий (6.19) она обращается в нуль. Если же эти условия нарушены (т. е. $g_i(\mathbf{x}) \neq 0$, $h_j(\mathbf{x}) < 0$ и $\operatorname{sign} h_j(\mathbf{x}) = -1$), то штрафная функция положительна. Она увеличивает вспомогательную целевую функцию $F(\mathbf{x}, \beta)$ тем больше, чем больше нарушаются условия (6.19).

При малых значениях параметра β вне области D функция $F(\mathbf{x}, \beta)$ сильно возрастает. Поэтому ее минимум может быть либо внутри D, либо снаружи вблизи границ этой области. В первом случае минимумы функций $F(\mathbf{x}, \beta)$ и $f(\mathbf{x})$ совпадают, поскольку дополнительные члены в (6.20) равны нулю. Если минимум функции $F(\mathbf{x}, \beta)$ находится вне D, то минимум целевой функции $f(\mathbf{x})$ лежит на границе D. При этом можно построить последовательность $\beta_k \to 0$ такую, что соответствующая последовательность минимумов функции $F(\mathbf{x}, \beta_k)$ будет стремиться к минимуму функции $f(\mathbf{x})$.

Таким образом, задача оптимизации для целевой функции $f(\mathbf{x})$ с ограничениями (6.19) свелась к последовательности задач безусловной оптимизации для вспомогательной функции (6.20), решение которых может быть проведено с помощью методов спуска. При этом строится итерационный процесс при $\beta \to 0$.

Алгоритм решения задачи математического программирования с использованием метода штрафных функций представлен в укрупненном виде на рис. 6.9. В качестве исходных данных вводятся начальное приближение искомого вектора $\mathbf{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$, начальное значение параметра β и некоторое малое число $\varepsilon > 0$, характеризующее точность расчета. На каждом шаге итерационного процесса определяется оптимальное значение $\mathbf{x}^*$ вектора $\mathbf{x}$, при этом в качестве начального приближения принимается результат предыдущей итерации. Значения параметра β каждый раз уменьшаются до тех пор, пока значение штрафной функции не станет меньше заданной малой величины. В этом случае точка $\mathbf{x}^*$ достаточно близка к границе области D и с необходимой точностью описывает оптимальные значения проектных параметров. Если точка минимума находится внутри области D, то искомый результат будет получен сразу после первого шага, поскольку в данном случае $\varphi(\mathbf{x}^*) = 0$.

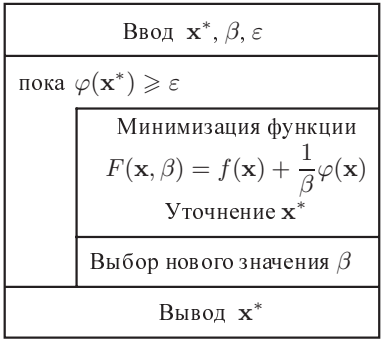


Рис. 6.9. Метод штрафных функций

2. Линейное программирование. До сих пор при рассмотрении задач оптимизации мы не делали никаких предположений о характере целевой функции и виде ограничений. Важным разделом математического программирования является *линейное программирование*, изучающее задачи оптимизации, в которых, целевая функция является линейной функцией проектных параметров, а ограничения задаются в виде линейных уравнений и неравенств.

Стандартная (каноническая) постановка задачи линейного программирования формулируется следующим образом: найти значения перемен-

ных $x_1, x_2, \ldots, x_n$, которые:

1) удовлетворяют системе линейных уравнений

$$\begin{aligned} a_{11}x_1 &+ a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 &+ a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 &+ a_{m2}x_2 + \ldots + a_{mn}x_n = b_m; \end{aligned} \tag{6.21}$$

2) являются неотрицательными, т. е.

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \ \ldots, \ x_n \geqslant 0; \tag{6.22}$$

3) обеспечивают наименьшее значение линейной целевой функции

$$f(x_1, x_2, \ldots, x_n) = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n. \tag{6.23}$$

Всякое решение системы уравнений (6.21), удовлетворяющее системе неравенств (6.22), называется *допустимым решением*. Допустимое решение, которое минимизирует целевую функцию (6.23), называется *оптимальным решением*.

Рассмотрим пример задачи линейного программирования (транспортную задачу).

Пример. Автобаза обслуживает три овощных магазина, причем товар доставляется в магазины из двух плодоовощных баз. Нужно спланировать перевозки так, чтобы их общая стоимость была минимальной.

Зададим исходные данные. Ежедневно вывозится с первой базы 12 т товара, со второй 15 т. При этом завозится в первый магазин 8 т, во второй 9 т, в третий 10 т. Стоимость перевозки 1 т товара (в рублях) с баз в магазины дается следующей таблицей:

База	Магазин		
	Первый	Второй	Третий
Первая	8	11	9
Вторая	10	7	12

Решение. Обозначим через x_1, x_2, x_3 количество товара, который нужно доставить с первой базы соответственно в первый, второй и третий магазины, а через x_4, x_5, x_6 количество товара, который нужно доставить со второй базы в те же магазины. Эти значения в соответствии с исходными данными должны удовлетворять следующим условиям:

$$\begin{aligned} x_1 + x_2 + x_3 &= 12, \\ x_4 + x_5 + x_6 &= 15, \\ x_1 + x_4 &= 8, \\ x_2 + x_5 &= 9, \\ x_3 + x_6 &= 10. \end{aligned} \tag{6.24}$$

Первые два уравнения этой системы описывают количество товара, которое необходимо вывезти с первой и второй баз, а три последних — сколько нужно завезти товара в каждый магазин.

К данной системе уравнений нужно добавить систему неравенств

$$x_i \geqslant 0, \quad i = 1, 2, \ldots, 6, \tag{6.25}$$

которая означает, что товар обратно с магазинов на базы не вывозится. Общая стоимость перевозок с учетом приведенных в таблице расценок выразится формулой

$$f = 8x_1 + 11x_2 + 9x_3 + 10x_4 + 7x_5 + 12x_6. \tag{6.26}$$

Таким образом, мы пришли к типичной задаче линейного программирования: найти оптимальные значения проектных параметров x_i ($i = 1, 2, \ldots, 6$), удовлетворяющих условиям (6.24), (6.25) и минимизирующих общую стоимость перевозок (6.26).

Из анализа системы уравнений (6.24) следует, что только первые четыре уравнения являются независимыми, а последнее можно получить из них (путем сложения первого и второго уравнений и вычитания из этой суммы третьего и четвертого уравнений). Поэтому фактически имеем систему

$$\begin{aligned} x_1 + x_2 + x_3 &= 12, \\ x_4 + x_5 + x_6 &= 15, \\ x_1 + x_4 &= 8, \\ x_2 + x_5 &= 9. \end{aligned}$$

Число неизвестных на два больше числа уравнений, поэтому выразим через x_1 и x_2 все остальные неизвестные. Получим

$$\begin{aligned} x_3 &= 12 - x_1 - x_2, \\ x_4 &= 8 - x_1, \\ x_5 &= 9 - x_2, \\ x_6 &= x_1 + x_2 - 2. \end{aligned} \tag{6.27}$$

Поскольку в соответствии с (6.25) все проектные параметры должны быть неотрицательны, то с учетом (6.27) получим следующую систему неравенств:

$$\begin{gathered} x_1 \geqslant 0, \quad x_2 \geqslant 0, \\ 12 - x_1 - x_2 \geqslant 0, \\ 8 - x_1 \geqslant 0, \quad 9 - x_2 \geqslant 0, \\ x_1 + x_2 - 2 \geqslant 0. \end{gathered} \tag{6.28}$$

Эти неравенства можно записать в более компактном виде:

$$0 \leqslant x_1 \leqslant 8, \quad 0 \leqslant x_2 \leqslant 9, \quad 2 \leqslant x_1 + x_2 \leqslant 12. \tag{6.29}$$

Данная система неравенств описывает все допустимые решения рассматриваемой задачи. Среди всех допустимых значений свободных параметров x_1 и x_2 нужно найти оптимальные, минимизирующие целевую функцию f. Формула (6.26) для нее с учетом соотношений (6.27) принимает вид

$$f = 227 + x_1 + 7x_2. \tag{6.30}$$

Отсюда следует, что стоимость перевозок растет с увеличением значений x_1, x_2; поэтому нужно взять их наименьшие допустимые значения. В соответствии с (6.29) $x_1 + x_2 \geqslant 2$; примем $x_1 + x_2 = 2$. Исключая один из параметров, например, x_2, получим $x_2 = 2 - x_1$. Тогда

$$f = 241 - 6x_1.$$

Очевидно, что стоимость перевозок будет минимальной, если величина x_1 примет наибольшее значение в рамках сделанного ограничения ($x_1 + x_2 \geqslant 2$). Таким оптимальным будет значение $x_1 = 2$. Тогда $x_2 = 0$, а оптимальные значения остальных проектных параметров можно найти по формулам (6.27): $x_3 = 10$, $x_4 = 6$, $x_5 = 9$, $x_6 = 0$. В этом случае минимальная общая стоимость перевозок равна 229 р. На рис. 6.10 пока-

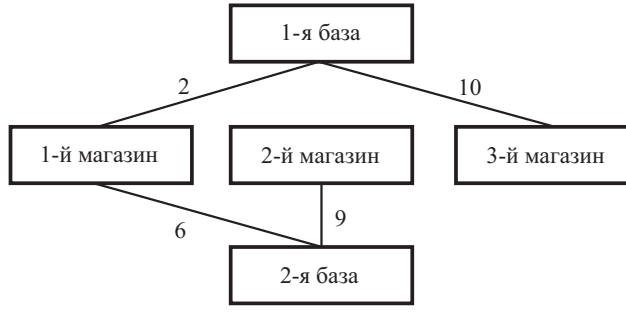


Рис. 6.10. Схема перевозок

зана схема доставки товаров, соответствующая полученному решению. Числа указывают количество товара (в тоннах).

3. Геометрический метод. Областью решения линейного неравенства с двумя переменными

$$a_0 + a_1x_1 + a_2x_2 \geqslant 0 \tag{6.31}$$

является полуплоскость. Для того чтобы определить, какая из двух полуплоскостей соответствует этому неравенству, нужно привести его к виду $x_2 \geqslant kx_1 + b$ или $x_2 \leqslant kx_1 + b$. Тогда искомая полуплоскость в первом случае расположена выше прямой $a_0 + a_1x_1 + a_2x_2 = 0$, во втором — ниже нее. Если $a_2 = 0$, то неравенство (6.31) имеет вид $a_0 + a_1x_1 \geqslant 0$; в этом случае получим либо $x \geqslant h$ — правую полуплоскость, либо $x \leqslant h$ — левую полуплоскость.

Областью решений системы неравенств является пересечение конечного числа полуплоскостей, описываемых каждым отдельным неравенством вида (6.31). Это пересечение представляет собой многоугольную область G. Она может быть как ограниченной, так и неограниченной и даже пустой (если система неравенств противоречива).

Область решений G обладает важным свойством выпуклости. Область называется *выпуклой*, если произвольные две ее точки можно соединить отрезком, целиком принадлежащим данной области.

На рис. 6.11 показаны выпуклая область G_1 и невыпуклая область G_2. В области G_1 две ее произвольные точки A_1 и B_1 можно соединить отрезком,

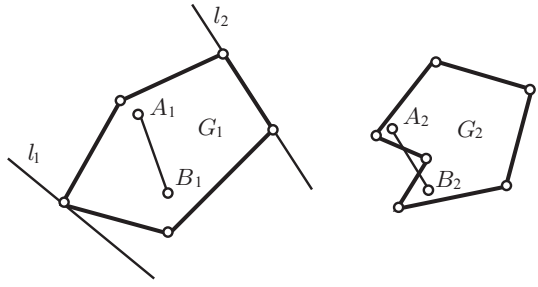


Рис. 6.11. Выпуклая (G_1) и невыпуклая (G_2) области

все точки которого принадлежат области G_1. В области G_2 можно выбрать такие две ее точки A_2 и B_2, что не все точки отрезка A_2B_2 принадлежат области G_2.

Опорной прямой называется прямая, которая имеет с областью по крайней мере одну общую точку, при этом вся область расположена по одну сторону от этой прямой.

На рис. 6.11 показаны две опорные прямые l_1 и l_2, т. е. в данном случае опорные прямые проходят соответственно через вершину многоугольника и через одну из его сторон.

Аналогично можно дать геометрическую интерпретацию системы неравенств с тремя переменными. В этом случае каждое неравенство описывает полупространство, а вся система — пересечение полупространств, т. е. многогранник, который также обладает свойством выпуклости. Здесь опорная плоскость проходит через вершину, ребро или грань многогранной области.

Основываясь на введенных понятиях, рассмотрим *геометрический метод* решения задачи линейного программирования. Пусть заданы линейная целевая функция $f = c_0 + c_1x_1 + c_2x_2$ двух независимых переменных, а также некоторая совместная система линейных неравенств, описывающих область решений G. Требуется среди допустимых решений $(x_1, x_2) \in G$ найти такое, при котором линейная целевая функция f принимает наименьшее значение.

Положим функцию f равной некоторому постоянному значению C: $f =$

$= c_0 + c_1 x_1 + c_2 x_2 = C$. Это значение достигается в точках прямой, удовлетворяющих уравнению

$$c_0 + c_1 x_1 + c_2 x_2 = C. \tag{6.32}$$

При параллельном переносе этой прямой в положительном направлении вектора нормали $\mathbf{n} = \{c_1, c_2\}$ линейная функция f будет возрастать, а при переносе прямой в противоположном направлении — убывать.

Действительно, пусть при параллельном переносе точка (x_1^*, x_2^*), принадлежащая прямой (6.32), переходит в точку $(x_1^* + \Delta x_1, x_2^* + \Delta x_2)$, принадлежащую новой прямой, т. е. параллельный перенос производится в направлении вектора $\Delta\mathbf{x} = \{\Delta x_1, \Delta x_2\}$. Тогда уравнение новой прямой будет иметь вид

$$c_0 + c_1 x_1 + c_2 x_2 = C + c_1 \Delta x_1 + c_2 \Delta x_2,$$

поскольку

$$c_0 + c_1 x_1 + c_2 x_2 =$$

$$= C_1 = c_0 + c_1(x_1^* + \Delta x_1) + c_2(x_2^* + \Delta x_2) = C + c_1 \Delta x_1 + c_2 \Delta x_2.$$

Если вектор $\Delta\mathbf{x}$ сонаправлен с вектором $\mathbf{n}$, то $\mathbf{n} \cdot \Delta\mathbf{x} = c_1 \Delta x_1 + c_2 \Delta x_2 > 0$ и $C_1 > C$, а если направлен противоположно, то $C_1 < C$.

Предположим, что прямая, записанная в виде (6.32), при параллельном переносе в положительном направлении вектора $\mathbf{n}$ первый раз встретится с областью допустимых решений G в некоторой ее вершине, при этом значение целевой функции равно C_1, и прямая становится опорной. Тогда значение C_1 будет минимальным, поскольку дальнейшее движение прямой в том же направлении приведет к увеличению значения f.

Если в задаче оптимизации нас интересует максимальное значение целевой функции, то параллельный перенос прямой (6.32) осуществляется в направлении, противоположном $\mathbf{n}$, пока она не станет опорной. Тогда вершина многоугольника, через которую проходит опорная прямая, будет соответствовать максимуму функции f. При дальнейшем переносе прямой целевая функция будет убывать.

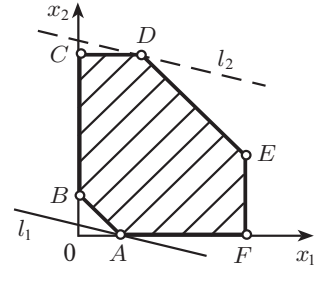


Рис. 6.12. Область допустимых решений

Таким образом, оптимизация линейной целевой функции на многоугольнике допустимых решений происходит в точках пересечения этого многоугольника с опорными прямыми, соответствующими данной целевой функции. При этом пересечение может быть в одной точке (в вершине многоугольника) либо в бесконечном множестве точек (на ребре многоугольника). В последнем случае имеется бесконечное множество оптимальных решений.

В заключение вернемся к рассмотренной ранее транспортной задаче (см. п. 2). На рис. 6.12 изображен многоугольник $ABCDEF$ допустимых

решений. Он получен как пересечение полуплоскостей, описываемых неравенствами (6.28). Опорная прямая l_1 соответствует уравнению (6.30) при $f = 229$. Точка A пересечения опорной прямой с многоугольником решений дает минимум целевой функции.

При дальнейшем параллельном переносе этой прямой вверх можем попасть в точку D (опорная прямая l_2) и получить максимум целевой функции.

4. Симплекс-метод. Рассмотренный геометрический метод решения задач линейного программирования достаточно прост и нагляден для случая двух и даже трех переменных. Для большего числа переменных применение геометрического метода становится невозможным.

Правда, мы видели, что оптимальные значения целевой функции достигаются на границе области допустимых решений. Поэтому в случае n неизвестных ($n > 3$) можно построить n-мерный многогранник решений, найти его вершины и вычислить значения целевой функции в этих точках. Наименьшее среди полученных значений можно принять за искомое, а координаты соответствующей вершины — за оптимальные значения проектных параметров.

Однако решение задачи линейного программирования не так просто, как может показаться на первый взгляд. Сложность состоит в том, что количество проектных параметров в реальных задачах (особенно в экономических) может достигать сотен и даже тысяч. При этом число вершин многогранника G может быть настолько большим, что перебор вершин и вычисление в них значений целевой функции приведет к такому объему вычислений, который практически невозможно осуществить в течение разумного времени даже с помощью компьютера.

Одним из методов, позволяющих эффективно решать подобные задачи, причем с гораздо меньшим числом операций, является симплекс-метод.

Симплексом называется простейший выпуклый многогранник при данном числе измерений. В частности, при $n = 2$ — произвольный треугольник, $n = 3$ — произвольный тетраэдр.

Идея *симплекс-метода* состоит в следующем. Примем в качестве начального приближения координаты некоторой вершины многогранника допустимых решений и найдем все ребра, выходящие из этой вершины. Двигаемся вдоль того ребра, по которому линейная целевая функция убывает. Приходим в новую вершину, находим все выходящие из нее ребра, двигаемся по одному из них и т. д. В конце концов мы придем в такую вершину, движение из которой вдоль любого ребра приведет к возрастанию целевой функции. Следовательно, минимум достигнут, и координаты этой последней вершины принимаются в качестве оптимальных значений рассматриваемых проектных параметров.

Отметим, что (поскольку f — линейная функция, а многогранник выпуклый) данный вычислительный процесс сходится к решению задачи, причем за конечное число шагов k. В данном случае их число порядка n, т. е. значительно меньше числа шагов в методе простого перебора вершин, где k может быть порядка 2^n.

Пусть задача линейного программирования состоит в том, что нужно найти такие неотрицательные значения проектных параметров x_1, $x_2, \ldots, x_n$, которые минимизируют линейную целевую функцию

$$f = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n. \tag{6.33}$$

при заданных ограничениях в виде системы линейных уравнений

$$\begin{aligned} a_{11}x_1 &+ a_{12}x_2 &+ \ldots &+ a_{1n}x_n &= b_1,\\ a_{21}x_1 &+ a_{22}x_2 &+ \ldots &+ a_{2n}x_n &= b_2,\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ a_{m1}x_1 &+ a_{m2}x_2 &+ \ldots &+ a_{mn}x_n &= b_m. \end{aligned} \tag{6.34}$$

Если в качестве ограничений заданы неравенства, то их можно преобразовать в равенства путем введения дополнительных неотрицательных переменных, которые называются *балансовыми*. Например, имея некоторое неравенство $a_1x_1 + a_2x_2 + \ldots + a_nx_n \leqslant b$, можно всегда ввести новую переменную $x_{n+1} \geqslant 0$, такую, что после ее прибавления к левой части данного неравенства получим равенство $a_1x_1 + a_2x_2 + \ldots + a_nx_n + x_{n+1} = b$.

Будем считать, что все уравнения системы (6.34) *линейно независимы*, т. е. ни одно из них нельзя получить как линейную комбинацию других. В противном случае линейно зависимые уравнения можем отбросить. И, кроме того, эта система совместна, т. е. среди уравнений системы нет противоречивых. Ясно, что при этих предположениях число уравнений m меньше числа переменных n, поскольку при $m = n$ система (6.34) имеет единственное решение, что исключает всякую оптимизацию; при $m > n$ не выполняются сделанные выше предположения.

Выразим первые m переменных $x_1, x_2, \ldots, x_m$ через остальные с помощью уравнений (6.34):

$$\begin{aligned} x_1 &= p_1 + q_{1,\,m+1}x_{m+1} + q_{1,\,m+2}x_{m+2} + \ldots + q_{1n}x_n,\\ x_2 &= p_2 + q_{2,\,m+1}x_{m+1} + q_{2,\,m+2}x_{m+2} + \ldots + q_{2n}x_n,\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ x_m &= p_m + q_{m,\,m+1}x_{m+1} + q_{m,\,m+2}x_{m+2} + \ldots + q_{mn}x_n. \end{aligned} \tag{6.35}$$

Здесь $p_i \geqslant 0$ $(i = 1, 2, \ldots m)$, поскольку при $x_{m+1} = x_{m+2} = \ldots = x_n = 0$ из (6.35) следует, что $x_i = p_i$, но $x_i \geqslant 0$.

Переменные x_1, x_2, $\ldots$, x_m называются *базисными*, а вектор $\{x_1, x_2, \ldots, x_m\}$ — *базисом*, x_{m+1}, x_{m+2}, $\ldots$, x_n — *свободные* переменные.

Используя соотношения (6.35), можно выразить линейную целевую функцию (6.33) через свободные переменные:

$$f = d_0 + d_{m+1}x_{m+1} + d_{m+2}x_{m+2} + \ldots + d_nx_n. \tag{6.36}$$

Процесс оптимизации начнем с некоторого начального (*опорного*) решения, например при нулевых значениях свободных переменных. Тогда получим

$$x_1 = p_1, \quad \ldots, \quad x_m = p_m, \quad x_{m+1} = 0, \quad \ldots, \quad x_n = 0. \tag{6.37}$$

При этом целевая функция (6.31) принимает значение $f^{(0)} = d_0$.

Дальнейшее решение задачи симплекс-методом распадается на ряд этапов, заключающихся в том, что от одного решения нужно перейти к другому с таким условием, чтобы целевая функция не возрастала. Это достигается выбором нового базиса и значений свободных переменных.

Выясним, является ли опорное решение (6.37) оптимальным. Для этого проверим, можно ли уменьшить соответствующее этому решению значение целевой функции $f = d_0$ при изменении каждой свободной переменной. Поскольку $x_i \geqslant 0$, то мы можем лишь увеличивать их значения. Если коэффициенты $d_{m+1}, \ldots, d_n$ в формуле (6.36) неотрицательны, то при увеличении любой свободной переменной $x_{m+1}, \ldots, x_n$ целевая функция не может уменьшиться. В этом случае решение (6.37) окажется оптимальным.

Пусть теперь среди коэффициентов формулы (6.36) хотя бы один отрицательный, например $d_{m+1} < 0$. Это означает, что при увеличении переменной x_{m+1} до некоторого значения $x_{m+1}^{(1)}$ целевая функция уменьшается по сравнению со значением d_0, соответствующим решению (6.37). Поэтому в качестве нового опорного выбирается решение при следующих значениях свободных параметров:

$$x_{m+1} = x_{m+1}^{(1)}, \quad x_{m+2} = 0, \ldots, x_n = 0.$$

При этом базисные переменные, вычисляемые по формулам (6.35), равны

$$x_i = p_i + q_{i,\,m+1} x_{m+1}^{(1)}, \quad i = 1, 2, \ldots, m. \tag{6.38}$$

Выясним теперь, как выбрать $x_{m+1}^{(1)}$. Если все коэффициенты $q_{i,\,m+1}$ неотрицательны, то x_{m+1} можно увеличивать неограниченно; в этом случае не существует оптимального решения задачи. Однако на практике такие случаи, как правило, не встречаются. Обычно среди коэффициентов $q_{i,\,m+1}$ имеются отрицательные, а это влечет за собой угрозу сделать некоторые переменные x_i в (6.38) отрицательными из-за большого значения $x_{m+1}^{(1)}$. Следовательно, переменную x_{m+1} можно увеличивать лишь до тех пор, пока базисные переменные остаются неотрицательными. Это и является условием выбора значения $x_{m+1}^{(1)}$. Его можно записать в виде

$$p_i + q_{i,\,m+1} x_{m+1}^{(1)} \geqslant 0, \quad i = 1, 2, \ldots, m. \tag{6.39}$$

Среди всех отрицательных коэффициентов $q_{i,\,m+1}$ найдем такой, для которого отношение $p_i/q_{i,\,m+1}$ является наименьшим по модулю. Пусть это элемент $q_{j,\,m+1}$. Обозначим его значение через Q, а соответствующее ему значение p_j через P. Тогда из (6.39) получим максимально возможное

значение переменной x_{m+1} на данном шаге оптимизации: $x_{m+1}^{(1)} = -P/Q$ ($P > 0$, $Q < 0$), и новое опорное решение запишем в виде

$$\begin{aligned} &x_i = p_i - \frac{P}{Q}\, q_{i,\,m+1}, \quad i = 1, \ldots, j-1, j+1, \ldots, m, \\ &x_j = 0, \quad x_{m+1} = -\frac{P}{Q}, \quad x_{m+2} = 0, \quad \ldots, \quad x_n = 0. \end{aligned} \tag{6.40}$$

Новая целевая функция при этих значениях проектных параметров равна

$$f^{(1)} = d_0 - d_{m+1}\frac{P}{Q}\,.$$

Полученное значение целевой функции $f^{(1)}$ меньше предыдущего, поскольку в данной формуле второй член правой части больше нуля ($d_{m+1} < 0$, $P > 0, Q < 0$).

На этом заканчивается первый шаг оптимизации. Теперь нужно сделать второй шаг, используя аналогичную процедуру. Для этого необходимо выбрать новый базис, принимая в качестве базисных переменных параметры $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{m+1}$. Переменные $x_j, x_{m+2}, \ldots, x_n$, принимающие нулевые значения, будут являться свободными. После второго шага мы либо найдем новые оптимальные значений переменных и соответствующее им значение целевой функции $f^{(2)} < f^{(1)}$, либо покажем, что решение (6.40) является оптимальным. В любом случае после конечного числа шагов мы придем к оптимальному решению. Еще раз подчеркнем, что в отличие от метода перебора симплекс-метод дает возможность вести поиск целенаправленно, уменьшая на каждом шаге значение целевой функции.

В качестве примера, иллюстрирующего симплекс-метод, рассмотрим задачу об использовании ресурсов.

5. Задача о ресурсах. В распоряжении бригады имеются следующие ресурсы: 300 кг металла, 100 м^2 стекла, 160 чел.-ч. (человеко-часов) рабочего времени. Бригаде поручено изготовлять два наименования изделий: А и Б. Цена одного изделия А 1 тыс. р., для его изготовления необходимо 4 кг металла, 2 м^2 стекла и 2 чел.-ч. рабочего времени. Цена одного изделия Б 1.2 тыс. р., для его изготовления необходимо 5 кг металла, 1 м^2 стекла и 3 чел.-ч. рабочего времени. Требуется так спланировать объем выпуска продукции, чтобы ее стоимость была максимальной.

Сначала сформулируем задачу математически. Обозначим через x_1 и x_2 количество изделий А и Б, которое необходимо запланировать (т. е. это искомые величины). Имеющиеся ресурсы сырья и рабочего времени зададим в виде ограничений-неравенств:

$$\begin{aligned} &4x_1 + 5x_2 \leqslant 300, \\ &2x_1 + \ \ x_2 \leqslant 100, \\ &2x_1 + 3x_2 \leqslant 160. \end{aligned} \tag{6.41}$$

Полная стоимость запланированной к производству продукции выражается

формулой

$$f = x_1 + 1.2\,x_2. \tag{6.42}$$

Таким образом, мы имеем задачу линейного программирования, которая состоит в определении оптимальных значений проектных параметров x_1, x_2 являющихся целыми неотрицательными числами, удовлетворяющих линейным неравенствам (6.41) и дающих максимальное значение линейной целевой функции (6.42).

Вид сформулированной задачи не является каноническим, поскольку условия (6.41) имеют вид неравенств, а не уравнений. Как уже отмечалось выше, такая задача может быть сведена к канонической путем введения дополнительных переменных x_3, x_4, x_5 по количеству ограничений-неравенств (6.41). При этом выбирают эти переменные такими, чтобы при их прибавлении к левым частям соотношений (6.41) неравенства превращались в равенства. Тогда ограничения примут вид

$$\begin{aligned} 4x_1 + 5x_2 + x_3 &= 300,\\ 2x_1 + \ x_2 + x_4 &= 100,\\ 2x_1 + 3x_2 + x_5 &= 160. \end{aligned} \tag{6.43}$$

При этом очевидно, что $x_3 \geqslant 0$, $x_4 \geqslant 0$, $x_5 \geqslant 0$. Заметим, что введение дополнительных неизвестных не повлияло на вид целевой функции (6.42), которая зависит только от параметров x_1, x_2. Фактически x_3, x_4, x_5 будут указывать остатки ресурсов, не использованные в производстве. Здесь мы имеем задачу максимизации, т. е. нахождения максимума целевой функции. Если функцию (6.42) взять со знаком минус и принять целевую функцию в виде

$$F = -x_1 - 1.2\,x_2. \tag{6.44}$$

то получим задачу минимизации для этой целевой функции.

Примем переменные x_3, x_4, x_5 в качестве базисных и выразим их через свободные переменные x_1, x_2 из уравнений (6.43). Получим

$$\begin{aligned} x_3 &= 300 - 4x_1 - 5x_2,\\ x_4 &= 100 - 2x_1 - \ x_2,\\ x_5 &= 160 - 2x_1 - 3x_2. \end{aligned} \tag{6.45}$$

В качестве опорного решения возьмем такое, которое соответствует нулевым значениям свободных параметров:

$$x_1^{(0)} = 0, \quad x_2^{(0)} = 0, \quad x_3^{(0)} = 300, \quad x_4^{(0)} = 100, \quad x_5^{(0)} = 160.$$

Этому решению соответствует нулевое значение целевой функции (6.44):

$$F^{(0)} = 0. \tag{6.46}$$

Исследуя полученное решение, отмечаем, что оно не является оптимальным, поскольку значение целевой функции (6.44) может быть уменьшено по сравнению с (6.46) путем увеличения свободных параметров.

Положим $x_2 = 0$ и будем увеличивать переменную x_1 до тех пор, пока базисные переменные остаются положительными. Из (6.45) следует, что x_1 можно увеличить до значения $x_1 = 50$, поскольку при большем его значении переменная x_4 станет отрицательной (отношение $100/(-2)$ является наименьшим по модулю среди отношений $300/(-4), 100/(-2), 160/(-2)$).

Таким образом, полагая $x_1 = 50$, $x_2 = 0$, получаем новое опорное решение (значения переменных x_3, x_4, x_5 найдем по формулам (6.45)):

$$x_1^{(1)} = 50, \quad x_2^{(1)} = 0, \quad x_3^{(1)} = 100, \quad x_4^{(1)} = 0, \quad x_5^{(1)} = 60. \tag{6.47}$$

Значение целевой функции (6.44) при этом будет равно

$$F^{(1)} = -50. \tag{6.48}$$

Новое решение (6.47), следовательно, лучше, поскольку значение целевой функции уменьшилось по сравнению с (6.46).

Следующий шаг начнем с выбора нового базиса. Примем ненулевые переменные в (6.47) x_1, x_3, x_5 в качестве базисных, а нулевые переменные x_2, x_4 в качестве свободных. Из системы (6.43) найдем

$$\begin{aligned} x_1 &= 50 - 0.5x_2 - 0.5x_4, \\ x_3 &= 100 - 3x_2 + 2x_4, \\ x_5 &= 60 - 2x_2 + x_4. \end{aligned} \tag{6.49}$$

Выражение для целевой функций (6.44) запишем через свободные параметры, заменив x_1 с помощью (6.49). Получим

$$F = -50 - 0.7x_2 + 0.5x_4. \tag{6.50}$$

Отсюда следует, что значение целевой функции по сравнению с (6.48) можно уменьшить за счет увеличения x_2 поскольку коэффициент при этой переменной в (6.50) отрицательный. При этом увеличение x_4 недопустимо, поскольку это привело бы к возрастанию целевой функции; поэтому положим $x_4 = 0$.

Максимальное значение переменной x_2 определяется соотношениями (6.49). Быстрее всех нулевого значения достигнет переменная x_5 при $x_2 = 30$. Дальнейшее увеличение x_2 поэтому невозможно. Следовательно, получаем новое опорное решение, соответствующее значениям $x_2 = 30$, $x_4 = 0$ и определяемое соотношениями (6.49):

$$x_1^{(2)} = 35, \quad x_2^{(2)} = 30, \quad x_3^{(2)} = 10, \quad x_4^{(2)} = 0, \quad x_5^{(2)} = 0. \tag{6.51}$$

При этом значение целевой функции (6.50) равно

$$F^{(2)} = -71.$$

Покажем, что полученное решение является оптимальным. Для проведения следующего шага ненулевые переменные в (6.51), т. е. x_1, x_2, x_3, нужно принять в качестве базисных, а нулевые переменные x_4, x_5 — в качестве свободных переменных. В этом случае целевую функцию можно записать в виде

$$F = -71 + 0.15\,x_4 + 0.35\,x_5.$$

Поскольку коэффициенты при x_4, x_5 положительные, то при увеличении этих параметров целевая функция возрастает. Следовательно, минимальное значение целевой функции $F_{\min} = -71$ соответствует нулевым значениям параметров x_4, x_5, и полученное решение является оптимальным.

Таким образом, ответ на поставленную задачу об использовании ресурсов следующий: для получения максимальной суммарной стоимости продукции при заданных ресурсах необходимо запланировать изготовление изделий А в количестве 35 штук и изделий Б в количестве 30 штук. Суммарная стоимость продукции равна 71 тыс. р. При этом все ресурсы стекла и рабочего времени будут использованы, а металла останется 10 кг.

Упражнения

1. Исследовать на экстремум функцию $y = (x-5)e^x$.

2. Найти наибольшее и наименьшее значения функции $y = x\sqrt{1-x^2}$ в области ее определения.

3. Удельный расход газа плотности ρ с показателем адиабаты k в газовой струе определяется формулой

$$Q = \rho v \left(1 - \frac{v^2}{v_{\max}^2}\right)^{1/(k-1)}.$$

При какой скорости v расход газа будет максимальным?

4. Записать алгоритм определения наименьшего значения функции на отрезке с помощью метода общего поиска.

5. Усовершенствовать алгоритм предыдущей задачи путем повторного деления суженного интервала неопределенности.

6. Используя метод золотого сечения, найти на отрезке $[0, 3]$ наименьшее значение функции

$$f(x) = \begin{cases} x^2 - 2x + 2, & 0 \leqslant x \leqslant 2, \\ x^2/(2x-1), & x > 2. \end{cases}$$

7. Работа деформации рамы выражается формулой

$$A = \frac{l^3}{2EI}\left(\frac{4}{3}X^2 - XY + \frac{1}{3}Y^2 + \frac{1}{3}PX - \frac{1}{4}PY + \frac{1}{10}P^2\right),$$

где P — нагрузка, X и Y — горизонтальная и вертикальная реакции опоры, l — длина, E — модуль упругости, I — момент инерции. При каких значениях X, Y работа будет минимальной?

8. Записать алгоритм решения задачи одномерной оптимизации методом Ньютона.

9*. Привести пример негладкой функции двух переменных, для которой метод покоординатного спуска не сойдется к точке минимума этой функции.

10. Записать алгоритм решения задачи многомерной оптимизации методом наискорейшего спуска.

11. Спроектировать цилиндрический котел емкостью 200 л таким образом, чтобы на его изготовление было израсходовано как можно меньше материала.

12. Начертить области, определенные системами неравенств:
а) $x \geqslant 0, \quad y \geqslant 0, \quad 2x + y \leqslant 4$;
б) $x - y \geqslant 0, \quad x \leqslant 9, \quad x + 3y \geqslant 6$.

13. Минимизировать функцию $f = 12x_1 + 4x_2$ при наличии ограничений $x_1 + x_2 \geqslant 2, \quad x_1 \geqslant 0.5, \quad x_2 \leqslant 4, \quad x_1 - x_2 \geqslant 0$.

14. Имеются два склада с сырьем. Ежедневно вывозится с первого склада 60 т сырья, со второго 80 т. Сырье используется двумя заводами, причем первый завод получает его 50 т, второй 90 т. Нужно организовать оптимальную (наиболее дешевую) схему перевозок, если известно, что доставка 1 т сырья с первого склада на первый завод стоит 7 р., с первого склада на второй завод — 9 р., со второго склада на первый завод — 10 р., со второго склада на второй завод — 8 р.

15*. Записать алгоритм решения задачи линейного программирования симплекс – методом.

ГЛАВА 7

ОБЫКНОВЕННЫЕ ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ

§ 1. Основные понятия

1. Постановка задач. Инженеру-исследователю постоянно приходится в своей деятельности сталкиваться с дифференциальными уравнениями. Многие задачи механики, физики, химии и других отраслей науки и техники при их математическом моделировании сводятся к дифференциальным уравнениям. В связи с этим решение дифференциальных уравнений является одной из важнейших математических задач. В вычислительной математике изучаются численные методы решения дифференциальных уравнений, которые особенно эффективны в сочетании с использованием вычислительной техники.

Прежде чем обсуждать методы решения дифференциальных уравнений, напомним некоторые сведения из курса дифференциальных уравнений, и в особенности те, которые понадобятся при дальнейшем изложении.

В зависимости от числа независимых переменных дифференциальные уравнения делятся на две существенно различные категории: обыкновенные дифференциальные уравнения, содержащие одну независимую переменную, и уравнения с частными производными, содержащие несколько независимых переменных. Данная глава посвящена методам решения обыкновенных дифференциальных уравнений.

Обыкновенными дифференциальными уравнениями называются такие уравнения, которые содержат одну или несколько производных от искомой функций $y = y(x)$. Их можно записать в виде

$$F(x, y, y', \ldots, y^{(n)}) = 0, \tag{7.1}$$

где x — независимая переменная.

Наивысший порядок n входящей в уравнение (7.1) производной называется *порядком дифференциального уравнения*. В частности, запишем уравнения первого и второго порядков:

$$F(x, y, y') = 0, \quad F(x, y, y', y'') = 0.$$

В ряде случаев из общей записи дифференциального уравнения (7.1) удается выразить старшую производную в явном виде. Например,

$$\begin{aligned} y' &= f(x, y), \\ y'' &= f(x, y, y'). \end{aligned} \tag{7.2}$$

Такая форма записи называется *уравнением, разрешенным относительно старшей производной*.

Линейным дифференциальным уравнением называется уравнение, линейное относительно искомой функции и ее производных. Например, $y' - x^2 y = \sin x$ — линейное уравнение первого порядка.

Решением дифференциального уравнения (7.1) называется всякая n раз дифференцируемая функция $y = \varphi(x)$, которая после ее подстановки в уравнение превращает его в тождество.

Общее решение обыкновенного дифференциального уравнения n-го порядка (7.1) содержит n произвольных постоянных $C_1, C_2, \ldots, C_n$:

$$y = \varphi(x, C_1, C_2, \ldots, C_n), \tag{7.3}$$

где (7.3) является решением уравнения (7.1) при любых значениях C_1, $C_2, \ldots, C_n$, а любое решение уравнения (7.1) можно представить в виде (7.3) при некоторых $C_1, C_2, \ldots, C_n$.

Частное решение дифференциального уравнения получается из общего, если произвольным постоянным придать определенные значения.

Для уравнения первого порядка общее решение зависит от одной произвольной постоянной:

$$y = \varphi(x, C). \tag{7.4}$$

Если постоянная принимает определенное значение $C = C_0$, то получается частное решение

$$y = \varphi(x, C_0).$$

Дадим геометрическую интерпретацию дифференциального уравнения первого порядка (7.2). Поскольку производная y' характеризует наклон касательной к графику решения $y = y(x)$ (*интегральной кривой*) в данной точке, то при $y' = k = \text{const}$ из (7.2) получим $f(x, y) = k$ — уравнение линии постоянного наклона, называемой *изоклиной*. Меняя k, получаем, семейство изоклин.

Приведем геометрическую интерпретацию общего решения (7.4). Это решение описывает бесконечное семейство интегральных кривых с параметром C, а частному решению соответствует одна кривая из этого семейства. При некоторых дополнительных предположениях через каждую точку (x_0, y_0) проходит одна и только одна интегральная кривая. Это утверждение следует из следующей теоремы.

Теорема Коши. *Если правая часть $f(x, y)$ уравнения* (7.2) *и ее частная производная $f'_y(x, y)$ определены и непрерывны в некоторой области G изменения переменных x, y, то для всякой внутренней точки (x_0, y_0) этой области данное уравнение имеет единственное решение, принимающее заданное значение $y = y_0$ при $x = x_0$.*

Для уравнений высших порядков геометрическая интерпретация более сложная. Через каждую точку в области решения уравнения при $n > 1$ проходит не одна интегральная кривая. Поэтому если для выделения некоторого частного решения уравнения первого порядка достаточно задать

координаты (x_0, y_0) произвольной точки на данной интегральной кривой, то для уравнений высших порядков этого недостаточно. Здесь правило следующее: для выделения частного решения из общего нужно задавать столько дополнительных условий, сколько произвольных постоянных в общем решении, т. е. каков порядок уравнения. Следовательно, для уравнения второго порядка нужно задать два дополнительных условия, благодаря которым можно найти значения двух произвольных постоянных.

В зависимости от способа задания дополнительных условий для получения частного решения дифференциального уравнения существуют два различных типа задач: задача Коши и краевая задача. В качестве дополнительных условий могут задаваться значения искомой функции и ее производных при некоторых значениях независимой переменной, т. е. в некоторых точках.

Если эти условия задаются в одной точке, то такая задача называется *задачей Коши*. Дополнительные условия в задаче Коши называются *начальными условиями*, а точка $x = x_0$, в которой они задаются, — *начальной точкой*.

Для уравнения первого порядка дополнительное условие одно, поэтому в этом случае может быть сформулирована только задача Коши: для заданных x_0, y_0 найти такое решение $y = y(x)$ уравнения (7.2), что $y(x_0) = y_0$. Таким образом, теорема Коши дает достаточные условия существования и единственности решения задачи Коши.

Если же для уравнения порядка $n > 1$ дополнительные условия задаются в более чем одной точке, т. е. при разных значениях независимой переменной, то такая задача называется *краевой*. Сами дополнительные условия называются при этом *граничными* (или *краевыми*) *условиями*. На практике обычно граничные условия задаются в двух точках $x = a$ и $x = b$, являющихся границами отрезка, на котором рассматривается дифференциальное уравнение.

Приведем примеры постановки задач для обыкновенных дифференциальных уравнений. Задачи Коши:

$$dx/dt = x^2 \cos t, \quad t > 0, \quad x(0) = 1;$$

$$y'' = y'/x + x^2, \quad x > 1, \quad y(1) = 2, \quad y'(1) = 0.$$

Краевые задачи:

$$y'' + 2y' - y = \sin x, \quad 0 \leqslant x \leqslant 1, \quad y(0) = 1, \quad y(1) = 0;$$

$$y''' = x + yy', \quad 1 \leqslant x \leqslant 3, \quad y(1) = 0, \quad y'(1) = 1; \quad y'(3) = 2.$$

2. О методах решения. Методы решения обыкновенных дифференциальных уравнений можно разбить на следующие группы: графические, аналитические, приближенные и численные.

Графические методы используют геометрические построения. В частности, одним из них является *метод изоклин* для решения дифференциальных

уравнений первого порядка вида (7.2). Он основан на геометрическом определении интегральных кривых по заранее построенному полю направлений, определенному изоклинами.

С некоторыми *аналитическими методами* читатель знаком по курсу дифференциальных уравнений. Для ряда уравнений первого порядка (с разделяющимися переменными, однородных, линейных и др.), а также для некоторых типов уравнений высших порядков (например, линейных с постоянными коэффициентами) удается получить решения в виде формул путем аналитических преобразований.

Приближенные методы используют различные упрощения самих уравнений путем обоснованного отбрасывания некоторых содержащихся в них членов, а также специальным выбором классов искомых функций. Например, в некоторых инженерных задачах удается представить решение в виде суммы двух составляющих, первое из которых определяет основное решение, а второе — малая добавка (*возмущение*), квадратом которой можно пренебречь. На этом основаны различные методы линеаризации. В приближенных методах также широко используется разложение решения в ряд по некоторому малому параметру, содержащемуся в данной задаче. К данной группе методов относятся и асимптотические методы, с помощью которых получаются решения, описывающие предельную картину рассматриваемого явления.

Здесь мы будем рассматривать численные методы решения дифференциальных уравнений, которые в настоящее время являются основным инструментом при исследовании научно-технических задач, описываемых дифференциальными уравнениями. При этом необходимо подчеркнуть, что данные методы особенно эффективны в сочетании с использованием современных компьютеров.

Наиболее распространенным и универсальным численным методом решения дифференциальных уравнений является *метод конечных разностей*. Его сущность состоит в следующем. Область непрерывного изменения аргумента (например, отрезок) заменяется дискретным множеством точек, называемых *узлами*. Эти узлы составляют *разностную сетку*. Искомая функция непрерывного аргумента приближенно заменяется функцией дискретного аргумента на заданной сетке. Эта функция называется *сеточной*. Исходное дифференциальное уравнение заменяется разностным уравнением относительно сеточной функции. При этом для входящих в уравнение производных используются соответствующие конечно-разностные соотношения (см. гл. 3, § 1).Такая замена дифференциального уравнения разностным называется его *аппроксимацией* на сетке (или *разностной аппроксимацией*). Таким образом, решение дифференциального уравнения сводится к отысканию значений сеточной функции в узлах сетки.

Обоснованность замены дифференциального уравнения разностным, точность получаемых решений, устойчивость метода — важнейшие вопросы, которые требуют тщательного изучения. Мы здесь дадим лишь некоторые элементарные сведения по данным вопросам.

3. Разностные методы. Обычно в теории разностных схем для компактности записи дифференциальные уравнения, начальные и граничные условия представляются в некотором символическом виде, называемом *операторным*. Например, любое из уравнений

$$Y' = f(x), \quad Y'' = f(x), \quad Y'' + k^2 Y = f(x)$$

можно записать в виде $LY = F(x)$. Здесь L — дифференциальный оператор, содержащий операции дифференцирования; его значение различно для разных дифференциальных уравнений. Область изменения аргумента x можно обозначить через G, т. е. $x \in G$. В частности, областью G при решении обыкновенных дифференциальных уравнений может быть некоторый отрезок $[a, b]$, полуось $x > 0$ (или $t > 0$) и т. п.

Дополнительные условия на границе также представляются в операторном виде. Например, любое из условий

$$Y(0) = A, \quad Y(a) = 0, \quad Y(b) = 1, \quad Y'(0) = B, \quad Y'(a) = 1$$

можно записать в виде $lY = \Phi(x)$ $(x \in \Gamma)$. Здесь l — оператор начальных или граничных условий, $\Phi(x)$ — правая часть этих условий, Γ — граница рассматриваемой области (т. е. точки $x = 0$, $x = a$, $x = b$ и т. п.).

Таким образом, исходную задачу для дифференциального уравнения с заданными начальными и граничными условиями, называемую в дальнейшем *дифференциальной задачей*, можно в общем случае записать в виде

$$LY = F(x), \quad x \in G, \tag{7.5}$$

$$lY = \Phi(x), \quad x \in \Gamma. \tag{7.6}$$

В методе конечных разностей исходное дифференциальное уравнение (7.5) заменяется разностным уравнением путем аппроксимации производных соответствующими конечно-разностными соотношениями. При этом в области G введем сетку, *шаг* $h > 0$ которой для простоты будем считать постоянным. Совокупность узлов x_0, x_1, ... обозначим через g_h. Значения искомой функции Y в узлах сетки заменяются значениями сеточной функции y_h, которая является решением разностного уравнения.

Искомую функцию и сеточную функцию будем обозначать соответственно Y и y, чтобы подчеркнуть их различие: Y — функция непрерывно меняющегося аргумента x, а y — дискретная сеточная функция, определенная на дискретном множестве $g_h = \{x_i\}$ $(i = 0, 1, \dots)$. Сеточную функцию, принимающую значения y_i в узлах сетки, можно считать функцией целочисленного аргумента i. Итак, дифференциальное уравнение (7.5) заменяется разностным уравнением, которое также можно записать в операторном виде:

$$L_h y_h = f_h, \quad x \in g_h. \tag{7.7}$$

Здесь L_h — разностный оператор, аппроксимирующий дифференциальный оператор L. Как известно (см. гл. 3, § 1), погрешность аппроксимации производных, а следовательно, и погрешность аппроксимации (7.7) в некоторой точке x может быть представлена в виде $\varepsilon(x) = O(h^k)$. При этом говорят, что в данной точке x имеет место *аппроксимация k-го порядка*. Индекс h в разностном уравнении (7.7) подчеркивает, что величина шага является параметром разностной задачи. Поэтому (7.7) можно рассматривать как целое семейство разностных уравнений, которые зависят от параметра h.

При решении дифференциальных уравнений обычно требуется оценить погрешность аппроксимации не в одной точке, а на всей сетке g_h, т. е. в точках x_0, x_1, ... В качестве погрешности аппроксимации ε_h на сетке можно принять некоторую величину, связанную с погрешностями аппроксимации в узлах например,

$$\varepsilon_h = \max_i |\varepsilon(x_i)|, \quad \varepsilon_h = \Big[\sum_i \varepsilon^2(x_i)\Big]^{1/2}.$$

В этом случае L_h имеет k-й порядок аппроксимации на сетке, если $\varepsilon_h = O(h^k)$.

Наряду с аппроксимацией (7.7) дифференциального уравнения (7.5) необходимо также аппроксимировать дополнительные условия на границе (7.6). Эти условия запишутся в виде

$$l_h y_h = \varphi_h, \quad x \in \gamma_h. \tag{7.8}$$

Здесь γ_h — множество граничных узлов сетки, т. е. $\gamma_h \subset \Gamma$. Индекс h, как и в (7.7), означает зависимость разностных условий на границе от значения шага.

Совокупность разностных уравнений (7.7), (7.8), аппроксимирующих исходное дифференциальное уравнение и дополнительные условия на границе, называется *разностной схемой*.

Пример. Рассмотрим задачу Коши

$$LY = \frac{dY}{dx} = F(x), \quad x > x_0, \quad Y(x_0) = A.$$

Введем равномерную сетку с шагом h, приняв в качестве узлов значения аргумента x_0, x_1, ... Значения сеточной функции, которая аппроксимирует искомое решение в данных узлах, обозначим через y_0, y_1, ... Тогда разностную схему можно записать, например, в виде

$$L_h y_h = \frac{y_{i+1} - y_i}{h} = f_i, \quad i = 0, 1, \ldots, \quad y_0 = A.$$

Здесь f_i — значение правой части разностного уравнения в точке x_i. Можно, в частности, принять $f_i = F(x_i)$. Данная схема имеет первый порядок аппроксимации, т. е. $\varepsilon_h = O(h)$.

Решение разностной задачи, в результате которого находятся значения сеточной функции y_i в узлах x_i, приближенно заменяет решение $Y(x)$ исходной дифференциальной задачи. Однако не всякая разностная схема дает

удовлетворительное решение, т. е. получаемые значения сеточной функции y_i не всегда с достаточной точностью аппроксимируют значения искомой функции $Y(x_i)$ в узлах сетки. Здесь важную роль играют такие понятия, как устойчивость, аппроксимация и сходимость разностной схемы.

Под *устойчивостью* схемы понимается непрерывная зависимость ее решения от входных данных (коэффициентов уравнений, правых частей, начальных и граничных условий). Или, другими словами, малому изменению входных данных соответствует малое изменение решения. В противном случае разностная схема называется *неустойчивой*. Естественно, что для практических расчетов используются устойчивые схемы, поскольку входные данные обычно содержат погрешности, которые в случае неустойчивых схем приводят к неверному решению. Кроме того, в расчетах на компьютере погрешности возникают в процессе счета из-за округлений, а использование неустойчивых разностных схем приводит к недопустимому накоплению этих погрешностей.

Разностная схема называется *корректной*, если ее решение существует и единственно при любых входных данных, а также если эта схема устойчива.

При использовании метода конечных разностей необходимо знать, с какой точностью решение разностной задачи приближает решение исходной дифференциальной задачи. Рассмотрим погрешность δ_h, равную разности значений искомой функции в узлах сетки и сеточной функции, т. е. $\delta_h = Y_h - y_h$. Отсюда найдем $y_h = Y_h - \delta_h$. Подставляя это значение y_h в разностную схему (7.7), (7.8), получаем

$$\begin{aligned} L_h Y_h - L_h \delta_h &= f_h, \quad x \in g_h, \\ l_h Y_h - l_h \delta_h &= \varphi_h, \quad x \in \gamma_h. \end{aligned}$$

Отсюда

$$L_h \delta_h = R_h, \quad l_h \delta_h = r_h.$$

Здесь $R_h = L_h Y_h - f_h$ — погрешность аппроксимации (*невязка*) для разностного уравнения, а $r_h = l_h Y_h - \varphi_h$ — погрешность аппроксимации для разностного граничного условия.

Если ввести характерные значения R и r невязок R_h и r_h (например, взять их максимальные по модулю значения на сетке), то при $R = O(h^k)$ и $r = O(h^k)$ говорят, что разностная схема (7.7), (7.8) *имеет k-й порядок аппроксимации* на решении.

Введем аналогичным образом характерное значение δ погрешности решения δ_h. Тогда разностная схема сходится, если $\delta \to 0$ при $h \to 0$. Если при этом $\delta = O(h^k)$, то говорят, что разностная схема имеет точность *k-го порядка* или *сходится со скоростью* $O(h^k)$.

В теории разностных схем доказывается, что если разностная схема устойчива и аппроксимирует исходную дифференциальную задачу, то она сходится. Иными словами, *из устойчивости и аппроксимации разностной схемы следует ее сходимость*. Это позволяет свести трудную задачу изучения сходимости и оценки порядка точности разностной схемы к изучению

погрешности аппроксимации и устойчивости, что значительно легче. Вопросы исследования разностных схем изложены в специальной литературе (см. список литературы).

§ 2. Задача Коши

1. Общие сведения. Требуется найти функцию $Y = Y(x)$, удовлетворяющую уравнению

$$Y' = f(x, Y) \tag{7.9}$$

и принимающую при $x = x_0$ заданное значение Y_0:

$$Y(x_0) = Y_0. \tag{7.10}$$

При этом будем для определенности считать, что решение нужно получить для значений $x > x_0$.

Согласно теореме Коши (см. § 1, п. 1) решение $Y(x)$ задачи (7.9), (7.10) существует, единственно и является гладкой функцией, если правая часть $f(x, Y)$ уравнения (7.9), являющаяся функцией двух переменных x, Y, удовлетворяет некоторым условиям гладкости. Будем считать, что эти условия выполнены и существует единственное гладкое решение $Y(x)$.

Методы решения задачи (7.9), (7.10) распространяются и на случай систем уравнений вида (7.9), а к ним в свою очередь можно привести также уравнения высших порядков. Например, уравнение

$$Z'' = \varphi(Z', Z, x)$$

можно записать в виде системы уравнений относительно функций Y_1, Y_2:

$$\begin{aligned} Y_1' &= \varphi(Y_1, Y_2, x), \\ Y_2' &= Y_1, \end{aligned} \tag{7.11}$$

где $Y_1 = Z'$, $Y_2 = Z$.

Систему (7.11) можно записать с помощью одного векторного уравнения:

$$\mathbf{Y}' = \mathbf{f}(\mathbf{Y}, x). \tag{7.12}$$

Здесь

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} Z' \\ Z \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \varphi \\ Z \end{pmatrix}.$$

Таким образом, векторное уравнение (7.12) можно использовать для замены как системы уравнений, так и уравнения порядка выше первого.

Для решения задачи Коши (7.9), (7.10) будем использовать разностные методы. Введем последовательность точек $x_0, x_1, \ldots$ и шаги $h_i = x_{i+1} - x_i$ $(i = 0, 1, \ldots)$. В каждой точке x_i называемой *узлом*, вместо значений

функции $Y(x_i)$ вводятся числа y_i, аппроксимирующие точное решение Y на данном множестве точек. Функцию y, заданную в виде таблицы $\{x_i, y_i\}$ $(i = 0, 1, \dots)$, называют *сеточной функцией*.

Далее, аппроксимируя в точке (x_i, y_i) значение производной в уравнении (7.9) отношением конечных разностей, осуществляем переход от дифференциальной задачи (7.9), (7.10) относительно функции Y к разностной задаче относительно сеточной функции y. Будем считать, что для разностной аппроксимации производной используется *шаблон*, состоящий из $k+1$ узла: $x_{i-k+1}, x_{i-k+2}, \dots, x_i, x_{i+1}$, причем значения $y_{i-k+1}, y_{i-k+2}, \dots, y_i$ уже найдены. Тогда разностное уравнение для нахождения значения y_{i+1} можно записать в общем виде как

$$y_{i+1} = F(x_i, y_{i+1}, y_i, \dots, y_{i-k+1}, h_i, h_{i-1}, \dots, h_{i-k+1}), \quad i = 0, 1, \dots,$$
$$y_0 = Y_0. \tag{7.13}$$

Здесь зависимость аппроксимации от $x_{i-k+1}, x_{i-k+2}, \dots, x_i, x_{i+1}$ сведена к зависимости от x_i и шагов $h_i, h_{i-1}, \dots, h_{i-k+1}$; конкретное выражение для правой части F зависит от способа аппроксимации производной. Для каждого численного метода получается свой вид уравнения (7.13).

На основании анализа вида разностного уравнения можно провести некоторую классификацию численных методов решения задачи Коши для обыкновенных дифференциальных уравнений.

Если в правой части (7.13) отсутствует y_{i+1}, т. е. значение y_{i+1} явно вычисляется по k предыдущим значениям $y_i, y_{i-1}, \dots, y_{i-k+1}$, то разностная схема называется *явной*. При этом получается *k-шаговый метод*: $k = 1$ — *одношаговый*, $k = 2$ — двухшаговый и т. д., т. е. в одношаговых методах для вычисления y_{i+1} используется лишь одно ранее найденное значение на предыдущем шаге y_i, в многошаговых — многие из них.

Если в правую часть уравнения (7.13) входит искомое значение y_{i+1}, то решение этого уравнения усложняется. В таких методах, называемых *неявными*, приходится решать уравнение (7.13) относительно y_{i+1} с помощью итерационных методов.

2. Метод Эйлера. Простейшим численным методом решения задачи Коши для обыкновенного дифференциального уравнения является *метод Эйлера*. Рассмотрим уравнение (7.9) в окрестностях узлов $x = x_i$ $(i = 0, 1, \dots)$ и заменим в левой части производную Y' правой разностью (3.4). При этом значения функции Y узлах x_i заменим значениями сеточной функций y_i:

$$\frac{y_{i+1} - y_i}{h_i} = f(x_i, y_i). \tag{7.14}$$

Полученная аппроксимация дифференциального уравнения (7.9) имеет первый порядок, поскольку при замене (7.9) на (7.14) допускается погрешность $O(h_i)$ (см. гл. 3, § 1).

Будем считать для простоты узлы равноотстоящими, т. е. $h_i = x_{i+1} - x_i = h = \text{const}$ $(i = 0, 1, \dots)$. Тогда из равенства (7.14) получаем

$$y_{i+1} = y_i + hf(x_i, y_i), \quad i = 0, 1, \dots \tag{7.15}$$

Заметим, что из уравнения (7.9) следует

$$Y'(x_i) = f(x_i, Y(x_i)) = f(x_i, y_i).$$

Поэтому (7.15) представляет собой приближенное нахождение значения функции Y в точке x_{i+1} при помощи разложения в ряд Тейлора с отбрасыванием членов второго и более высоких порядков. Другими словами, приращение функции полагается равным ее дифференциалу.

Полагая $i = 0$, с помощью соотношения (7.15) находим значение сеточной функции y_1 при $x = x_1$:

$$y_1 = y_0 + hf(x_0, y_0).$$

Требуемое здесь значение y_0 задано начальным условием (7.10), т. е.

$$y_0 = Y_0. \tag{7.16}$$

Аналогично могут быть найдены значения сеточной функции в других узлах:

$$y_2 = y_1 + hf(x_1, y_1),$$
$$\dots\dots\dots\dots$$
$$y_n = y_{n-1} + hf(x_{n-1}, y_{n-1}),$$
$$\dots\dots\dots\dots$$

Построенный алгоритм называется методом Эйлера. Разностная схема этого метода представлена соотношениями (7.15), (7.16). Они имеют вид рекуррентных формул, с помощью которых значение сеточной функции y_{i+1} в любом узле x_{i+1} вычисляется по ее значению y_i в предыдущем узле x_i. В связи с этим метод Эйлера относится к одношаговым методам.

Структурограмма алгоритма решения задачи Коши (7.9), (7.10) методом Эйлера изображена на рис. 7.1. Задаются начальные значения $x = x_0$, $y = y_0$, а также величина шага h и количество расчетных точек n. Решение получается в узлах $x + h$, $x + 2h$, $\dots$, $x + nh$. Вывод результатов предусмотрен на каждом шаге. Если найденные значения необходимо хранить в памяти машины, то следует ввести массив значений $y_0, y_1, \dots, y_n$.

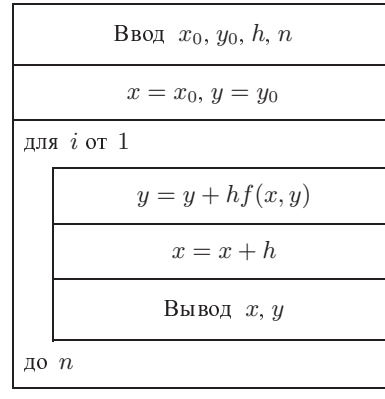


Рис. 7.1. Метод Эйлера

Приведенным алгоритмом можно воспользоваться и в случае, если требуется найти решение задачи Коши на отрезке $[a, b]$ при начальном условии в точке $x = a$. Нужно положить $x_0 = a$, $h = (b - a)/n$.

Геометрическая интерпретация метода Эйлера дана на рис. 7.2. Изображены первые два шага, т. е. проиллюстрировано вычисление сеточной функции в точках x_1, x_2. Интегральные кривые 0, 1, 2 описывают точные решения уравнения (7.9). При этом кривая 0 соответствует точному решению задачи Коши (7.9), (7.10), так как она проходит через начальную точку $A(x_0, y_0)$. Точки B, C получены в результате численного решения задачи Коши методом Эйлера. Их отклонения от кривой 0 характеризуют погрешность метода. При выполнении каждого шага мы фактически попадаем на другую интегральную кривую. Отрезок AB — отрезок касательной к кривой 0 в точке A, ее наклон характеризуется значением производной $Y'(x_0) = f(x_0, y_0)$. Погрешность появляется потому, что приращение значения функции при переходе от x_0 к x_1 заменяется приращением ординаты касательной к кривой 0 в точке A. Касательная BC уже проводится к другой интегральной кривой 1. Таким образом, погрешность метода Эйлера приводит к тому, что на каждом шаге приближенное решение переходит на другую интегральную кривую. Рассмотрим подробнее вопрос о погрешности метода Эйлера.

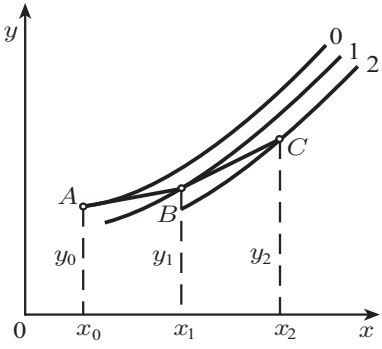


Рис. 7.2. Иллюстрация метода Эйлера

Погрешность δ_i в точке x_i равна разности между точным значением искомой функции $Y(x_i)$ и значением сеточной функции y_i: $\delta_i = Y(x_i) - y_i$. Выясним, чему будет равна погрешность при вычислении y_{i+1}. Для этого подставим $y_i = Y(x_i) - \delta_i$ и $y_{i+1} = Y(x_{i+1}) - \delta_{i+1}$ в (7.15). Имеем

$$Y(x_{i+1}) - \delta_{i+1} = Y(x_i) - \delta_i + hf(x_i,\, Y(x_i) - \delta_i). \qquad (7.17)$$

Разложим функцию f в ряд в окрестности точки $(x_i, Y(x_i))$:

$$f(x_i, Y(x_i) - \delta_i) = f(x_i, Y(x_i)) - \frac{\partial f}{\partial Y}\,\delta_i + O(\delta_i^2) = f(x_i, Y(x_i)) + O(\delta_i).$$

Используя полученное разложение, выразим δ_{i+1} из (7.17):

$$\delta_{i+1} = \delta_i + Y(x_{i+1}) - Y(x_i) - hf(x_i, Y(x_i)) + hO(\delta_i).$$

Учитывая, что $Y(x_{i+1}) = Y(x_i) + hf(x_i, Y(x_i)) + O(h^2)$, получаем

$$\delta_{i+1} = \delta_i + O(h^2) + hO(\delta_i). \qquad (7.18)$$

Таким образом, погрешность δ_{i+1} отличается от погрешности δ_i на два

слагаемых: $O(h^2)$ есть следствие погрешности аппроксимации (7.14), а $hO(\delta_i)$ есть следствие неточности значения y_i.

При нахождении y_1 начальное значение y_0 задается, как правило, точно: $\delta_0 = 0$. Отсюда

$$\delta_1 = O(h^2), \quad \delta_2 = \delta_1 + O(h^2) + hO(h^2) = \delta_1 + O(h^2) = O(h^2), \quad \dots$$

Мы видим, что последнее слагаемое в (7.18) можно отбросить:

$$\delta_{i+1} = \delta_i + O(h^2),$$

т. е. погрешность на каждом шаге увеличивается на величину $O(h^2)$.

При нахождении решения в точке x_n, отстоящей на конечном расстоянии L от точки x_0, погрешность состоит из n слагаемых $O(h^2)$. Если учесть, что $h = L/n$, то для погрешности δ_n получаем окончательное выражение:

$$\delta_n = nO(h^2) = \frac{L}{h}\,O(h^2) = O(h). \tag{7.19}$$

Таким образом, мы показали, что метод Эйлера имеет первый порядок точности.

3. Модификации метода Эйлера. Рассмотрим уравнение (7.9) в окрестностях узлов $x = x_i + h/2$ $(i = 0, 1, \dots)$, являющихся серединами отрезков $[x_i, x_{i+1}]$. В левой части (7.9) заменим производную центральной разностью (3.5), а в правой части заменим значение функции $f\big(x_i + h/2, Y(x_i + h/2)\big)$ средним арифметическим значений функции $f(x, Y)$ в точках (x_i, y_i) и (x_{i+1}, y_{i+1}). Тогда вместо (7.14) напишем

$$\frac{y_{i+1} - y_i}{h} = \frac{1}{2}\big[f(x_i, y_i) + f(x_{i+1}, y_{i+1})\big]. \tag{7.20}$$

Отсюда

$$y_{i+1} = y_i + \frac{h}{2}\big[f(x_i, y_i) + f(x_{i+1}, y_{i+1})\big]. \tag{7.21}$$

Полученная схема является неявной, поскольку искомое значение y_{i+1} входит в обе части соотношения (7.21) и его, вообще говоря, нельзя выразить явно. Для вычисления y_{i+1} можно применить один из итерационных методов. Если имеется хорошее начальное приближение y_i, то можно построить решение с использованием двух итераций следующим образом. Считая y_i начальным приближением, вычисляем первое приближение $\tilde{y}_{i+1}$ по формуле метода Эйлера (7.15):

$$\tilde{y}_{i+1} = y_i + hf(x_i, y_i). \tag{7.22}$$

Вычисленное значение $\tilde{y}_{i+1}$ подставляем вместо y_{i+1} в правую часть соотношения (7.21) и находим окончательное значение

$$y_{i+1} = y_i + \frac{h}{2}\big[f(x_i, y_i) + f(x_{i+1}, \tilde{y}_{i+1})\big]. \tag{7.23}$$

Алгоритм (7.22), (7.23) можно записать в виде одного соотношения:

$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f\big(x_{i+1}, y_i + hf(x_i, y_i)\big)\right], \quad i = 0, 1, \ldots$$

Данные рекуррентные соотношения описывают новую разностную схему, являющуюся модификацией метода Эйлера, которая называется *методом Эйлера с пересчетом*. Покажем, что этот метод отличается от метода Эйлера большей точностью. Аппроксимация (7.20) имеет, в отличие от (7.14), второй порядок. Действительно, при замене производной в левой части (7.9) допускается ошибка $O(h^2)$. Погрешность такого же порядка имеет место и при замене правой части (7.9) правой частью (7.20):

$$\frac{1}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right] = \frac{1}{2}\left[f\left(x_i + \frac{h}{2},\, Y\left(x_i + \frac{h}{2}\right)\right) - \frac{\partial f}{\partial x}\frac{h}{2} - \right.$$
$$- \frac{\partial f}{\partial Y}Y'\frac{h}{2} + O(h^2) + f\left(x_i + \frac{h}{2},\, Y\left(x_i + \frac{h}{2}\right)\right) + \frac{\partial f}{\partial x}\frac{h}{2} +$$
$$\left. + \frac{\partial f}{\partial Y}Y'\frac{h}{2} + O(h^2)\right] = f\left(x_i + \frac{h}{2},\, Y\left(x_i + \frac{h}{2}\right)\right) + O(h^2).$$

Здесь проведено разложение функции $f(x, Y)$ в ряд в окрестности точки $(x_i + h/2,\ Y(x_i + h/2))$.

Погрешность, допускаемая при вычислении y_{i+1} по формуле (7.21), составляет $hO(h^2) = O(h^3)$. Этот порядок погрешности сохраняется и при использовании двух итераций (7.22), (7.23), поскольку

$$f(x_{i+1}, \tilde{y}_{i+1}) =$$
$$= f(x_{i+1}, y_{i+1}) + \frac{\partial f}{\partial Y}(\tilde{y}_{i+1} - y_{i+1}) + O(h^2) = f(x_{i+1}, y_{i+1}) + O(h^2).$$

Таким образом, погрешность на каждом шаге (локальная) имеет порядок $O(h^3)$, а суммарная по аналогии с (7.19) — $O(h^2)$ в отличие от $O(h)$ в обычном методе Эйлера, т. е. метод Эйлера с пересчетом имеет второй порядок точности.

Заметим, что при использовании неявной схемы (7.21) получается практически то же значение y_{i+1}, что и в методе Эйлера с пересчетом. Однако применение схемы (7.21), требующей построения итерационного процесса для вычисления значения y_{i+1}, привело бы к значительному возрастанию времени счета на каждом шаге.

На рис. 7.3 дана геометрическая интерпретация первого шага вычислений при решении задачи Коши методом Эйлера с пересчетом. Касательная к кривой $Y(x)$ в точке (x_0, y_0) проводится с угловым коэффициентом $y' = f(x_0, y_0)$. С ее помощью методом Эйлера (7.15) найдено значение $\tilde{y}_1$, которое используется затем для определения наклона касательной $f(x_1, \tilde{y}_1)$ в точке (x_1, y_1). Отрезок с таким наклоном заменяет

первоначальный отрезок касательной от точки $x_0 + h/2$ до точки x_1. В результате получается уточненное значение искомой функции y_1 в этой точке.

С помощью метода Эйлера с пересчетом можно проводить контроль точности решения путем сравнения значений $\tilde{y}_{i+1}$ и y_{i+1} и выбора на основании этого соответствующей величины шага h в каждом узле. А именно, если величина $|y_{i+1} - \tilde{y}_{i+1}|$ сравнима с погрешностями вычислений, то шаг нужно увеличить; в противном случае, если эта разность слишком велика (например, $|y_{i+1} - \tilde{y}_{i+1}| > 0.01|y_{i+1}|$), значение h следует уменьшить. Используя эти оценки, можно построить алгоритм метода Эйлера с пересчетом с автоматическим выбором шага. Рекомендуем читателю составить такой алгоритм.

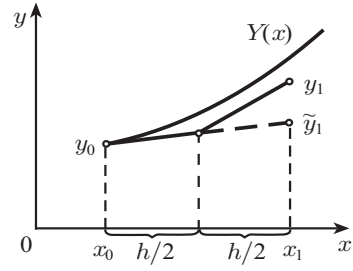


Рис. 7.3. Метод Эйлера с пересчетом

Наряду с методом Эйлера с пересчетом используется и другая модификация метода Эйлера. Так же, как и в методе Эйлера с пересчетом, рассмотрим уравнение (7.9) в окрестностях узлов $x = x_i + h/2$ $(i = 0, 1, \dots)$. В левой части (7.9) заменим производную центральной разностью (3.5), а правую часть оставим без изменений:

$$\frac{y_{i+1} - y_i}{h} = f\left(x_i + \frac{h}{2},\, Y\left(x_i + \frac{h}{2}\right)\right). \tag{7.24}$$

Приближенное значение функции Y в точке $(x_i + h/2)$ вычислим с помощью метода Эйлера:

$$\tilde{y} = y_i + \frac{h}{2} f(x_i, y_i). \tag{7.25}$$

Выразим y_{i+1} из (7.24), заменив $Y(x_i + h/2)$ его приближением $\tilde{y}$:

$$y_{i+1} = y_i + hf(x_i, \tilde{y}), \quad i = 0, 1, \dots \tag{7.26}$$

Полученный метод (7.25), (7.26) называется *усовершенствованным методом Эйлера*. Нетрудно показать, что он также имеет второй порядок точности.

4. Методы Рунге–Кутта. Существуют и другие явные одношаговые методы. Так, рассмотренные метод Эйлера (7.15) и его модифицированные варианты (7.22), (7.23) и (7.25), (7.26) являются частными случаями методов первого и второго порядков, относящихся к классу *методов Рунге–Кутта*. Эти методы используют для вычисления значения y_{i+1} $(i = 0, 1, \dots)$ значение y_i, а также значения функции $f(x, y)$ при некоторых специальным образом выбираемых значениях $x \in [x_i, x_{i+1}]$ и y. На их основе могут быть построены разностные схемы разного порядка точности.

Широко распространен метод Рунге – Кутта четвертого порядка.

Запишем алгоритм этого метода в виде

$$y_{i+1} = y_i + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3), \quad i = 0, 1, \ldots,$$
$$k_0 = f(x_i, y_i), \quad k_1 = f(x_i + \frac{h}{2},\ y_i + \frac{hk_0}{2}), \tag{7.27}$$
$$k_2 = f(x_i + \frac{h}{2},\ y_i + \frac{hk_1}{2}), \quad k_3 = f(x_i + h,\ y_i + hk_2).$$

Таким образом, данный метод Рунге–Кутта требует на каждом шаге четырехкратного вычисления правой части $f(x, Y)$ уравнения (7.9). Суммарная погрешность этого метода есть величина $O(h^4)$.

Метод Рунге–Кутта (7.27) требует большего объема вычислений по сравнению с методом Эйлера и его модификациями, однако это окупается повышенной точностью, что дает возможность проводить счет с бóльшим шагом. Другими словами, для получения результатов с одинаковой точностью в методе Эйлера потребуется значительно меньший шаг, чем в методе Рунге–Кутта (7.27).

Проведем сравнительную оценку рассмотренных методов Рунге–Кутта на простом примере, позволяющем получить также и точное решение.

Пример. Решить задачу Коши

$$Y' = 2(x^2 + Y), \quad Y(0) = 1, \quad 0 \leqslant x \leqslant 1, \quad h = 0.1.$$

Решение. Сформулированная задача Коши может быть решена известными из курса высшей математики методами. Опустив выкладки, запишем окончательное выражение для точного решения с учетом заданного начального условия. Оно имеет вид

$$Y(x) = 1.5\, e^{2x} - x^2 - x - 0.5.$$

Проведем теперь решение данной задачи численно с помощью рассмотренных выше методов. Результаты вычислений приведены в табл. 7.1. Анализ решения позволяет проследить рост погрешности с возрастанием x_i. Как видно из табл. 7.1, самым точным является решение, полученное методом Рунге–Кутта четвертого порядка. При $x_i = 1$ погрешность составляет менее $0.003\,\%$. Для модифицированных методов Эйлера погрешность при $x_i = 1$ составляет около $1\,\%$, а для самого метода Эйлера — почти $18\,\%$. Следовательно, при бóльших x метод Эйлера может привести к еще более существенным погрешностям, и в таких случаях предпочтительнее пользоваться численными методами высших порядков точности.

С уменьшением шага h локальная погрешность метода Эйлера снизится, однако при этом возрастет число узлов, что неблагоприятно повлияет на точность результатов. Поэтому метод Эйлера применяется сравнительно редко при небольшом числе расчетных точек. Наиболее употребительным одношаговым методом является метод Рунге–Кутта.

Таблица 7.1

x_i	Метод Эйлера	Метод Эйлера с пересчетом	Усовершенствованный метод Эйлера	Метод Рунге–Кутта 4-го порядка	Точное решение
0.1	1.2000	1.2210	1.2205	1.2221	1.2221
0.2	1.4420	1.4948	1.4937	1.4977	1.4977
0.3	1.7384	1.8375	1.8356	1.8432	1.8432
0.4	2.1041	2.2685	2.2658	2.2783	2.2783
0.5	2.5569	2.8118	2.8079	2.8274	2.8274
0.6	3.1183	3.4964	3.4912	3.5201	3.5202
0.7	3.8139	4.3578	4.3509	4.3927	4.3928
0.8	4.6747	5.4393	5.4304	5.4894	5.4895
0.9	5.7377	6.7938	6.7824	6.8643	6.8645
1.0	7.0472	8.4856	8.4713	8.5834	8.5836

Рассмотренные методы Рунге–Кутта могут быть использованы так же для решения систем дифференциальных уравнений. Покажем это для случая системы двух уравнений относительно искомых функций $Y=Y(x)$, $Z=Z(x)$ вида

$$Y' = \varphi(x, Y, Z),$$
$$Z' = \psi(x, Y, Z).$$

Начальные условия зададим в виде

$$Y(x_0) = Y_0, \quad Z(x_0) = Z_0.$$

По аналогии с (7.27) запишем формулы Рунге–Кутта для системы двух уравнений:

$$y_{i+1} = y_i + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3),$$
$$z_{i+1} = z_i + \frac{h}{6}(l_0 + 2l_1 + 2l_2 + l_3), \quad i = 0, 1, \ldots,$$

$$k_0 = \varphi(x_i, y_i, z_i), \quad l_0 = \psi(x_i, y_i, z_i),$$

$$k_1 = \varphi\left(x_i + \frac{h}{2},\ y_i + \frac{hk_0}{2},\ z_i + \frac{hl_0}{2}\right),$$

$$l_1 = \psi\left(x_i + \frac{h}{2},\ y_i + \frac{hk_0}{2},\ z_i + \frac{hl_0}{2}\right),$$

$$k_2 = \varphi\left(x_i + \frac{h}{2},\ y_i + \frac{hk_1}{2},\ z_i + \frac{hl_1}{2}\right),$$

$$l_2 = \psi\left(x_i + \frac{h}{2},\ y_i + \frac{hk_1}{2},\ z_i + \frac{hl_1}{2}\right),$$

$$k_3 = \varphi(x_i + h,\ y_i + hk_2,\ z_i + hl_2),$$

$$l_3 = \psi(x_i + h,\ y_i + hk_2,\ z_i + hl_2).$$

К решению систем уравнений сводятся также задачи Коши для уравнения высших порядков. Например, рассмотрим задачу Коши для уравнения второго порядка

$$Y'' = f(x, Y, Y'),$$
$$Y(x_0) = Y_0, \quad Y'(x_0) = Z_0.$$

Введем вторую неизвестную функцию $Z(x) = Y'(x)$. Тогда сформулированная задача Коши заменяется следующей:

$$Z' = f(x, Y, Z),$$
$$Y' = Z,$$
$$Y(x_0) = Y_0, \quad Z(x_0) = Z_0.$$

В заключение еще раз отметим особенность одношаговых методов, состоящую в том, что для получения решения в каждом новом расчетном узле достаточно иметь значение сеточной функции лишь в предыдущем узле. Это позволяет непосредственно начать счет при $i = 0$ по известным начальным значениям. Кроме того, указанная особенность допускает изменение шага в любой точке в процессе счета, что позволяет строить численные алгоритмы с автоматическим выбором шага.

5. Многошаговые методы. Другой путь построения разностных схем основан на том, что для вычисления значения y_{i+1} используются результаты не одного, а k предыдущих шагов, т. е. значения $y_{i-k+1}, y_{i-k+2}, \ldots, y_i$. В этом случае получается k-шаговый метод.

Многошаговые методы могут быть построены следующим образом. Запишем исходное уравнение (7.9) в виде

$$dY(x) = f(x, Y)dx. \tag{7.28}$$

Проинтегрируем обе части этого уравнения по x на отрезке $[x_i, x_{i+1}]$. Интеграл от левой части легко вычисляется:

$$\int_{x_i}^{x_{i+1}} dY(x) = Y(x_{i+1}) - Y(x_i) \approx y_{i+1} - y_i. \tag{7.29}$$

Для вычисления интеграла от правой части уравнения (7.28) строится сначала интерполяционный многочлен P_{k-1} степени $k-1$ для аппроксимации функции $f(x, Y)$ на отрезке $[x_i, x_{i+1}]$ по значениям

$f(x_{i-k+1}, y_{i-k+1})$, $f(x_{i-k+2}, y_{i-k+2})$, ..., $f(x_i, y_i)$. После этого можно написать

$$\int_{x_i}^{x_{i+1}} f(x, Y)\,dx \approx \int_{x_i}^{x_{i+1}} P_{k-1}(x)\,dx. \tag{7.30}$$

Приравнивая выражения, полученные в (7.29) и (7.30), можно получить формулу для определения неизвестного значения сеточной функции y_{i+1} в узле x_{i+1}:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} P_{k-1}(x)\,dx.$$

На основе этой формулы можно строить различные многошаговые методы любого порядка точности. Порядок точности зависит от степени интерполяционного многочлена $P_{k-1}(x)$, для построения которого используются значения сеточной функции y_i, y_{i-1}, ..., y_{i-k+1}, вычисленные на k предыдущих шагах.

Широко распространенным семейством многошаговых методов являются *методы Адамса*. Простейший из них, получающийся при $k = 1$, совпадает с рассмотренным ранее методом Эйлера первого порядка точности. В практических расчетах чаще всего используется вариант метода Адамса, имеющий четвертый порядок точности и использующий на каждом шаге результаты предыдущих четырех. Именно его и называют обычно методом Адамса. Рассмотрим этот метод.

Пусть найдены значения y_{i-3}, y_{i-2}, y_{i-1}, y_i в четырех последовательных узлах ($k = 4$). При этом имеются также вычисленные ранее значения правой части f_{i-3}, f_{i-2}, f_{i-1}, f_i, где $f_l = f(x_l, y_l)$. В качестве интерполяционного многочлена $P_3(x)$ можно взять многочлен Ньютона (см. гл. 2, § 3). В случае постоянного шага h конечные разности для правой части в узле x_i имеют вид

$$\begin{aligned}
\Delta f_i &= f_i - f_{i-1},\\
\Delta^2 f_i &= f_i - 2f_{i-1} + f_{i-2},\\
\Delta^3 f_i &= f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}.
\end{aligned}$$

Тогда разностную схему четвертого порядка метода Адамса можно записать после необходимых преобразований в виде

$$y_{i+1} = y_i + hf_i + \frac{h^2}{2}\,\Delta f_i + \frac{5h^3}{12}\,\Delta^2 f_i + \frac{3h^4}{8}\,\Delta^3 f_i. \tag{7.31}$$

Сравнивая метод Адамса с методом Рунге–Кутта той же точности, отмечаем его экономичность, поскольку он требует вычисления лишь одного значения правой части на каждом шаге (в методе Рунге–Кутта — четырех). Но метод Адамса неудобен тем, что невозможно начать счет по одному лишь известному значению y_0. Расчет может быть начат только

с узла x_3, а не x_0. Значения y_1, y_2, y_3, необходимые для вычисления y_4, нужно получить каким-либо другим способом (например, методом Рунге–Кутта), что существенно усложняет алгоритм. Кроме того, метод Адамса не позволяет (без усложнения формул) изменить шаг h в процессе счета; этого недостатка лишены одношаговые методы.

Рассмотрим еще одно семейство многошаговых методов, которые используют неявные схемы, — *методы прогноза и коррекции* (они называются также *методами предиктор-корректор*). Суть этих методов состоит в следующем. На каждом шаге вводятся два этапа, использующих многошаговые методы: с помощью явного метода (*предиктора*) по известным значениям функции в предыдущих узлах находится начальное приближение $y_{i+1} = y_{i+1}^{(0)}$ в новом узле; используя неявный метод (*корректор*), в результате итераций находятся приближения $y_{i+1}^{(1)}$, $y_{i+1}^{(2)}$, ...

Один из вариантов метода прогноза и коррекции может быть получена основе метода Адамса четвертого порядка. Приведем окончательный вид разностных соотношений: на этапе предиктора

$$y_{i+1} = y_i + \frac{h}{24}\,(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}); \qquad (7.32)$$

на этапе корректора

$$y_{i+1} = y_i + \frac{h}{24}\,(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}). \qquad (7.33)$$

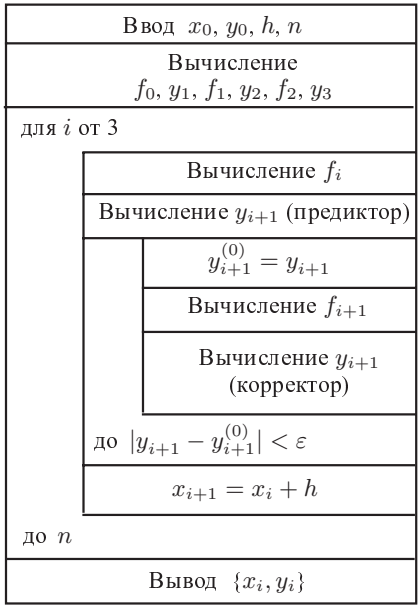


Рис. 7.4. Метод предиктор-корректор

Явная схема (7.32) используется на каждом шаге один раз, а с помощью неявной схемы (7.33) строится итерационный процесс вычисления y_{i+1}, поскольку это значение входит в правую часть выражения $f_{i+1} = f(x_{i+1}, y_{i+1})$.

Заметим, что в этих формулах, как и в случае метода Адамса, при вычислении y_{i+1} необходимы значения сеточной функции в четырех предыдущих узлах: y_i, y_{i-1}, y_{i-2}, y_{i-3}. Следовательно, расчет по этому методу может быть начат только со значения y_4. Необходимые при этом y_1, y_2, y_3 находятся по методу Рунге–Кутта, y_0 задается начальным условием. Это характерная особенность многошаговых методов. Алгоритм решения задачи Коши с помощью рассмотренного метода прогноза и коррекции представлен в укрупненном виде на рис. 7.4.

6. Повышение точности результатов. Точность численного решения можно повысить различными путями. В частности, этого можно достичь, применяя разностные схемы повышенного порядка точности. Однако такие схемы целесообразно строить лишь для уравнений с постоянными коэффициентами, поскольку в случае переменных коэффициентов схемы высоких порядков приводят к трудоемким алгоритмам.

Точность можно повысить также путем уменьшения значения шага h. Но и этот путь ограничен требованием экономичности, поскольку получение решения с необходимой точностью может потребовать огромного объема вычислений.

На практике часто для повышения точности численного решения без существенного увеличения машинного времени используется *метод Рунге*. Аналогичный метод для задачи численного дифференцирования был рассмотрен в п. 5 § 1 гл. 3. Метод Рунге состоит в том, что проводятся повторные расчеты по одной разностной схеме с различными шагами. Уточненное решение в совпадающих при разных расчетах узлах строится с помощью проведенной серии расчетов.

Предположим, что проведены две серии расчетов по схеме порядка k соответственно с шагами h и $h/2$. В результате расчетов получены множества значений сеточной функции y_h и $y_{h/2}$. Применим для уточнения решения формулу Рунге (3.16)[1)], положив в ней $k = 1/2$, заменив $F(x)$ на $Y(x)$, $f(x, h)$ на y_h, $f(x, kh)$ на $y_{h/2}$ и переобозначив порядок точности p через k. Для искомой функции $Y(x)$ в узлах сетки с шагом h имеем

$$Y = \frac{2^k y_{h/2} - y_h}{2^k - 1} + O(h^{k+1}). \tag{7.34}$$

Отбрасывая последнее слагаемое, получаем уточненное значение y_h^* сеточной функции в узлах сетки с шагом h:

$$y_h^* = \frac{2^k y_{h/2} - y_h}{2^k - 1}, \quad Y = y_h^* + O(h^{k+1}).$$

Порядок точности этого решения равен $k + 1$, хотя используемая разностная схема имеет порядок точности k. Таким образом, решение задачи на двух сетках позволяет на порядок повысить точность результатов.

Для схемы Эйлера первого порядка точности ($k = 1$) формула Рунге принимает вид

$$y_h^* = 2^k y_{h/2} - y_h, \quad Y = y_h^* + O(h^2).$$

Аналогично можно записать формулу для уточнения решения, полученного по методу Рунге–Кутта при $k = 4$.

[1)] Легко убедиться, что при выводе формулы (3.16) несущественно, какова природа аппроксимируемой функции; это может быть производная, решение задачи Коши и т. д.

Метод Рунге можно применить не только для уточнения решения, но и для оценки порядка точности разностной схемы тогда, когда он неизвестен. Для этого нужно провести расчет с тремя различными шагами h, $h/2$, $h/4$. Обозначим соответствующие сеточные функции, значения которых получены в расчетах, через y_h, $y_{h/2}$ и $y_{h/4}$. Тогда для значения порядка точности k, имеет место оценка

$$k \approx \log_2 \frac{y_h - y_{h/2}}{y_{h/2} - y_{h/4}}. \tag{7.35}$$

§ 3. Краевые задачи

1. Предварительные замечания. В § 2 рассматривались задачи с начальными условиями, т. е. с условиями в одной (начальной) точке: при $x = x_0$, $t = t_0$ и т. п. На практике приходится часто решать задачи другого типа, когда условия задаются при двух значениях независимой переменной (на концах рассматриваемого отрезка). Такие задачи, называемые *краевыми*, получаются при решении уравнений высших порядков или систем уравнений.

Рассмотрим, например, линейное дифференциальное уравнение второго порядка

$$Y'' + p(x)Y' + q(x)Y = f(x). \tag{7.36}$$

Краевая задача состоит в отыскании решения $Y = Y(x)$ уравнения (7.36) на отрезке $[a, b]$, удовлетворяющего на концах отрезка условиям

$$Y(a) = A, \quad Y(b) = B. \tag{7.37}$$

Граничные условия могут быть заданы не только в виде (7.37), но и в более общем виде:

$$\begin{aligned} \alpha_1 Y(a) + \beta_1 Y'(a) &= A, \\ \alpha_2 Y(b) + \beta_2 Y'(b) &= B. \end{aligned} \tag{7.38}$$

Методы решения краевых задач довольно разнообразные — это и точные аналитические методы, и приближенные, и численные (см. § 1, п. 2).

Аналитические методы изучаются в курсе дифференциальных уравнений. Они имеются лишь для решения узкого класса уравнений. В частности, хорошо развит этот аппарат для решения линейных дифференциальных уравнений второго порядка с постоянными коэффициентами, которые широко используются в исследовании различных физических процессов (например, в теории колебаний, динамике твердого тела и т. п.).

Приближенные методы разрабатывались еще задолго до появления компьютеров. Однако многие из них и до сих пор не утратили своего значения. Это методы коллокаций, наименьших квадратов и другие,

основанные на минимизации невязок уравнений. Весьма эффективными являются также метод Галеркина и его модификации. Рассмотрим сущность приближенных методов.

Для отыскания приближенного решения уравнения (7.36) с граничными условиями (7.38) выбирается некоторая линейно независимая (*базисная*) система дважды дифференцируемых функций $\varphi_0(x)$, $\varphi_1(x)$, ..., $\varphi_n(x)$. При этом $\varphi_0(x)$ удовлетворяет граничным условиям (7.38), а $\varphi_1(x)$, ..., $\varphi_n(x)$ — условиям (7.38) с нулевыми правыми частями A, B [1]. Искомое приближенное решение представляется в виде линейной комбинации базисных функций:

$$y(x) = \varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) + \ldots + a_n\varphi_n(x). \tag{7.39}$$

Подставляя это выражение в уравнение (7.36), можно найти разность между его левой и правой частями, которая называется *невязкой*. Она является функцией переменной x и параметров a_1, a_2, ..., a_n и имеет вид

$$\psi(x, a_1, a_2, \ldots, a_n) = y'' + p(x)y' + q(x)y - f(x). \tag{7.40}$$

Коэффициенты a_1, a_2, ..., a_n стараются подобрать так, чтобы невязка была в каком-то смысле минимальной. Способ определения этих коэффициентов и характеризует тот или иной приближенный метод.

В *методе коллокаций* выбираются n точек $x = x_i$ ($i = 1, 2, \ldots, n$, $x_i \in [a, b]$), называемых *точками коллокации*, невязки (7.40) в которых приравниваются нулю. Получается система n линейных алгебраических уравнений относительно a_1, a_2, ..., a_n. Решая данную систему, можно найти эти коэффициенты, которые затем подставляются в решение (7.39).

Метод наименьших квадратов основан на минимизации суммы квадратов невязок в заданной системе точек $x_1, x_2, \ldots, x_m$. Из этого условия также получается система линейных алгебраических уравнений относительно $a_1, a_2, \ldots, a_n$.

В основе *метода Галеркина* лежит требование ортогональности базисных функций $\varphi_1(x)$, $\varphi_2(x)$, ..., $\varphi_n(x)$ к невязке $\psi(x, a_1, \ldots, a_n)$, которое выражается в виде

$$\int_a^b \psi(x, a_1, \ldots, a_n)\varphi_i(x)dx = 0, \quad i = 1, 2, \ldots, n.$$

Из этих условий также получается система линейных алгебраических уравнений относительно коэффициентов линейной комбинации (7.39).

Аналогично строятся некоторые другие приближенные методы. Все они сводятся к построению системы линейных алгебраических уравнений, из которой, если существует ее решение, находятся неизвестные коэффициенты. Они затем используются для построения решения как линейной комбинации базисных функций.

---

[1] Такие условия называются *однородными*.

Дальше будут рассмотрены численные методы. Их можно разделить на две группы: сведение решения краевой задачи к последовательности решений задач Коши и непосредственное применение конечно-разностных методов.

2. Метод стрельбы. Рассмотрим краевую задачу для уравнения второго порядка, разрешенного относительно второй производной:

$$Y'' = f(x, Y, Y'). \tag{7.41}$$

Будем искать решение $Y = Y(x)$ этого уравнения на отрезке $[0, 1]$. Любой отрезок $[a, b]$ можно привести к этому отрезку с помощью замены переменной

$$t = \frac{x - a}{b - a}.$$

Граничные условия на концах рассматриваемого отрезка примем в простейшем виде (7.37), т. е.

$$Y(0) = Y_0, \quad Y(1) = Y_1. \tag{7.42}$$

Сущность *метода стрельбы* заключается в сведении решения краевой задачи (7.41), (7.42) к решению последовательности задач Коши для того же уравнения (7.41) с начальными условиями

$$Y(0) = Y_0, \quad Y'(0) = \operatorname{tg}\alpha. \tag{7.43}$$

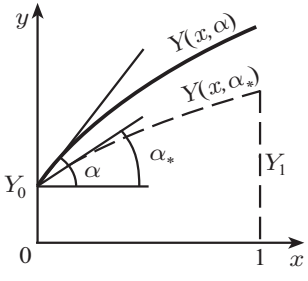


Рис. 7.5. Метод стрельбы

Здесь Y_0 — точка на оси ординат, в которой помещается начало искомой интегральной кривой; α — угол наклона касательной к интегральной кривой в этой точке (рис. 7.5).

Считая решение задачи Коши $Y{=}Y(x, \alpha)$ зависящим от параметра α, будем искать такую интегральную кривую $Y{=}Y(x, \alpha_*)$, которая выходит из точки $(0, Y_0)$ и попадает в точку $(1, Y_1)$. Таким образом, если $\alpha = \alpha_*$, то решение $Y(x, \alpha)$ задачи Коши совпадает с решением $Y(x)$ краевой задачи. При $x = 1$, учитывая второе граничное условие (7.42), получаем $Y(1, \alpha) = Y_1$, или

$$Y(1, \alpha) - Y_1 = 0. \tag{7.44}$$

Следовательно, для нахождения параметра α получим уравнение вида $F(\alpha) = 0$, где $F(\alpha) = Y(1, \alpha) - Y_1$. Это уравнение отличается от привычной записи тем, что функцию $F(x)$ нельзя представить в виде некоторого аналитического выражения, поскольку она является решением задачи Коши (7.41), (7.43). Тем не менее для решения уравнения (7.44) может быть использован любой из рассмотренных ранее методов решения нелинейных уравнений (см. гл. 5).

Например, при использовании метода деления отрезка пополам поступаем следующим образом. Находим начальный отрезок $[\alpha_0, \alpha_1]$, содержащий значение α_*, на концах которого функция $F(x)$ принимает значения разных знаков. Для этого решение задачи Коши $Y(x, \alpha_0)$ должно при $x = 1$ находиться ниже точки Y_1, а $Y(x, \alpha_1)$ — выше. Далее, полагая $\alpha_2 = (\alpha_0 + \alpha_1)/2$, снова решаем задачу Коши при $\alpha = \alpha_2$ и в соответствии с методом деления отрезка пополам отбрасываем один из отрезков: $[\alpha_0, \alpha_2]$ или $[\alpha_2, \alpha_1]$, на котором функция $F(x)$ не меняет знак, и т. д. (см. алгоритм на рис. 5.2). Процесс поиска решения прекращается, если разность двух последовательно найденных значений α меньше некоторого наперед заданного малого числа. В этом случае полученное последним решение задачи Коши и будет принято за искомое решение краевой задачи.

Описанный алгоритм называется *методом стрельбы* вполне оправданно, поскольку в нем как бы проводится «пристрелка» по углу наклона интегральной кривой в начальной точке. Следует отметить, что этот алгоритм хорошо работает в том случае, если решение $Y(x, \alpha)$ не слишком чувствительно к изменениям α; в противном случае мы можем столкнуться с неустойчивостью.

Для решения уравнения (7.44) используются и другие методы. В частности, одним из самых надежных является *метод Ньютона*. Его применение состоит в следующем. Пусть α_0 — начальное приближение к α_*. Построим итерационный процесс для нахождения последующих приближений α_k с помощью формулы метода Ньютона (5.11):

$$\alpha_k = \alpha_{k-1} - \frac{F(\alpha_{k-1})}{F'(\alpha_{k-1})}, \quad k = 1, 2, \ldots$$

С учетом того, что $F'(\alpha) = \partial Y(x, \alpha)/\partial\alpha$, имеем

$$\alpha_k = \alpha_{k-1} + \frac{Y_1 - Y(x, \alpha_{k-1})}{\partial Y(x, \alpha_{k-1})/\partial\alpha}, \quad k = 1, 2, \ldots \tag{7.45}$$

Производную в знаменателе этого выражения можно найти численно:

$$\frac{\partial Y(x, \alpha_{k-1})}{\partial\alpha} \approx \frac{Y(x, \alpha_{k-1} + \Delta\alpha) - Y(x, \alpha_{k-1})}{\Delta\alpha}. \tag{7.46}$$

Здесь $\Delta\alpha$ — произвольное малое возмущение α.

Для вычисления правой части (7.46) нужно решить задачу Коши при $\alpha = \alpha_{k-1} + \Delta\alpha$, в результате чего найдем значение $Y(x, \alpha_{k-1} + \Delta\alpha)$. Затем по формуле (7.45) находим следующее приближение α_k параметра α_* и т. д. Этот итерационный процесс продолжается до тех пор, пока два последовательных приближения α_{k-1} и α_k не станут отличаться меньше, чем на заданное малое число ε.

Алгоритм решения краевой задачи методом стрельбы с применением пристрелки по методу Ньютона представлен на рис. 7.6. Нахождение

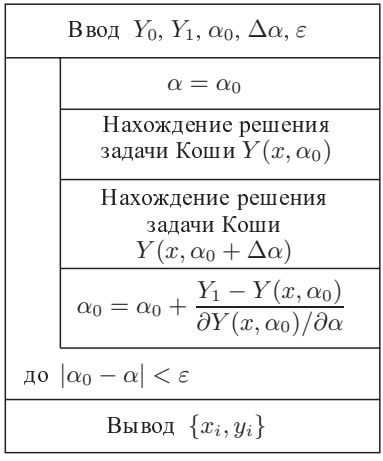


Рис. 7.6. Алгоритм метода стрельбы

решения задачи Коши $Y(x, \alpha)$ входит в данный алгоритм в качестве отдельного модуля с входным данным α. На выходе модуля получается решение $Y(x, \alpha)$ в виде значений y_i ($i = 0, 1, \ldots, n$) в точках $x_i = ih$, где $h = 1/n$.

Методы стрельбы могут также использоваться для решения системы уравнений. В этом случае краевая задача (а не задача Коши) может возникнуть в силу того, что значения одной части искомых функций заданы при одном значении независимой переменной (например, при $x = 0$), а другой — при другом (например, $x = 1$). Тогда «пристрелка» проводится по неизвестным значениям искомых функций при $x = 0$ до тех пор, пока не будут удовлетворяться соответствующие граничные условия при $x = 1$.

Например, рассмотрим систему двух уравнений первого порядка

$$\begin{aligned} Y' &= f_1(x, Y, Z), \\ Z' &= f_2(x, Y, Z). \end{aligned} \tag{7.47}$$

Граничные условия заданы в виде

$$Y(0) = Y_0, \quad Z(1) = Z_1. \tag{7.48}$$

Процесс решения этой краевой задачи методом стрельбы состоит в следующем. Выбирается некоторое α, являющееся начальным приближением для $Z(0)$. Решается задача Коши для системы (7.47) с начальными условиями $Y(0) = Y_0$, $Z(0) = \alpha$. В результате решения при $x = 1$ получается некоторое значение $Z(1, \alpha) \neq Z_1$. Если разность между этими величинами невелика, то найденное решение задачи Коши принимается за искомое решение краевой задачи. В противном случае находится уточненное значение α и процесс повторяется.

Таким образом, метод стрельбы может быть также использован как для решения краевых задач для уравнений высших порядков, так и для систем уравнений.

3. Методы конечных разностей. Достоинство этих методов состоит в том, что они сводят решение краевой задачи для дифференциального уравнения к решению системы алгебраических уравнений относительно значений искомой функции на заданном множестве точек. Это достигается путем замены производных, входящих в дифференциальное уравнение, их конечно-разностными аппроксимациями (см. гл. 3, § 1).

Рассмотрим сущность такого метода решения для дифференциального уравнения второго порядка (7.41) при заданных граничных условиях (7.42). Разобьем отрезок $[0,1]$ на n равных частей точками $x_i = ih$ $(i = 0, 1, \ldots, n)$. Решение краевой задачи (7.41), (7.42) сведем к вычислению значений сеточной функции y_i в узловых точках x_i. Для этого напишем уравнение (7.42) для внутренних узлов:

$$Y''(x_i) = f(x_i, Y(x_i), Y'(x_i)), \quad i = 1, 2, \ldots, n-1. \tag{7.49}$$

Заменим производные, входящие в эти соотношения, их конечно-разностными аппроксимациями:

$$Y'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2), \quad Y''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2). \tag{7.50}$$

Подставляя эти выражения в (7.49), получаем систему разностных уравнений

$$F(x_i, y_{i-1}, y_i, y_{i+1}) = 0, \quad i = 1, 2, \ldots, n-1, \tag{7.51}$$

являющуюся системой $n-1$ алгебраических уравнений относительно значений сеточной функции y_1, y_2, $\ldots$, y_{n-1}. Входящие в данную систему y_0 (при $i = 1$) и y_n (при $i = n-1$) берутся из граничных условий (7.42):

$$y_0 = Y_0, \quad y_n = Y_1.$$

На практике часто граничные условия задаются в более общем виде (7.38):

$$\begin{aligned} \alpha_1 Y(0) + \beta_1 Y'(0) &= A, \\ \alpha_2 Y(1) + \beta_2 Y'(1) &= B. \end{aligned} \tag{7.52}$$

В этом случае граничные условия также должны представляться в разностном виде путем аппроксимации производных $Y'(0)$ и $Y'(1)$ с помощью конечно-разностных соотношений. Если использовать односторонние разности (соответствующий шаблон показан на рис. 7.7, *а*), при которых производные аппроксимируются с первым порядком точности, то разностные граничные условия примут вид

$$\begin{aligned} \alpha_1 y_0 + \beta_1 \frac{y_1 - y_0}{h} &= A, \\ \alpha_2 y_n + \beta_2 \frac{y_n - y_{n-1}}{h} &= B. \end{aligned} \tag{7.53}$$

Из этих соотношений легко находятся значения y_0, y_n.

Однако, как правило, предпочтительнее аппроксимировать производные, входящие в (7.52), со вторым порядком точности с помощью центральных разностей

$$Y'(0) = \frac{y_1 - y_{-1}}{2h} + O(h^2), \quad Y'(1) = \frac{y_{n+1} - y_{n-1}}{2h} + O(h^2).$$

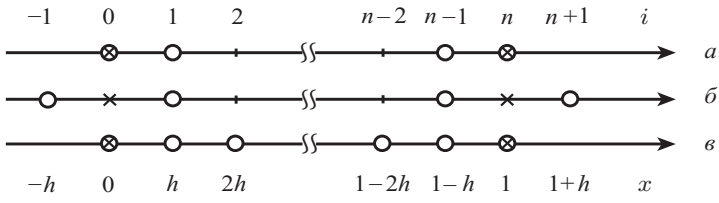


Рис. 7.7. Аппроксимация граничных условий

В данные выражения входят значения сеточной функции y_{-1} и y_{n+1} в так называемых *фиктивных узлах* $x = -h$ и $x = 1 + h$, лежащих вне рассматриваемого отрезка (см. рис. 7.7, *б*). В этих узлах значения искомой функции также должны быть найдены. Следовательно, количество неизвестных значений сеточной функции увеличивается на два. Для замыкания системы привлекают еще два разностных уравнения (7.51) при $i = 0$, $i = n$.

Аппроксимировать граничные условия со вторым порядком можно и иначе (см. рис. 7.7, *в*). В этом случае используются аппросимации, полученные в п. 3 § 1 гл. 3:

$$Y'(0) = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2), \quad Y'(1) = \frac{y_{n-2} - 4y_{n-1} + 3y_n}{2h} + O(h^2).$$

Таким образом, решение краевой задачи для дифференциального уравнения сведено к решению системы алгебраических уравнений вида (7.51). Эта система является линейной или нелинейной в зависимости от того, линейно или нелинейно дифференциальное уравнение (7.41). Методы решения таких систем рассмотрены ранее (см. гл. 4, 5).

Рассмотрим подробнее один частный случай, который представляет интерес с точки зрения практических приложений и позволяет проследить процесс построения разностной схемы. Решим краевую задачу для линейного дифференциального уравнения второго порядка

$$\begin{gathered} Y''(x) - p(x)Y(x) = f(x), \\ p(x) > 0, \quad 0 \leqslant x \leqslant 1. \end{gathered} \tag{7.54}$$

с граничными условиями вида

$$Y(0) = A, \quad Y(1) = B. \tag{7.55}$$

Разобьем отрезок $[0, 1]$ на части с постоянным шагом h с помощью узлов $x_i = ih$ $(i = 0, 1, \ldots, n)$. Аппроксимируем вторую производную Y'' конечно-разностным соотношением (7.50). При этом значения искомой функции в узлах $Y(x_i)$ приближенно заменяем соответствующими значениями сеточной функции y_i. Записывая уравнение (7.54) в каждом узле с использованием указанных аппроксимаций, получаем

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - p(x_i)y_i = f(x_i).$$

Обозначим через p_i, f_i соответственно значения $p(x_i)$, $f(x_i)$. После несложных преобразований приведем последнее равенство к виду

$$y_{i-1} - (2 + h^2 p_i) y_i + y_{i+1} = h^2 f_i, \quad i = 1, 2, \ldots, n-1. \tag{7.56}$$

Получилась система $n-1$ линейных уравнений, число которых совпадает с числом неизвестных значений сеточной функции y_1, y_2, ..., y_{n-1} в узлах. Ее значения на концах отрезка определены граничными условиями (7.55):

$$y_0 = A, \quad y_n = B. \tag{7.57}$$

Решая систему уравнений (7.56) с учетом условий (7.57), находим значения сеточной функции, которые приближенно равны значениям искомой функции. Покажем, что такое решение существует и сходится к точному решению при $h \to 0$.

Для доказательства существования решения рассмотрим систему линейных уравнений (7.56). Ее матрица является трехдиагональной; на главной диагонали находятся элементы $-(2 + h^2 p_i)$. Поскольку $p(x) > 0$, то $p_i > 0$, и диагональные элементы матрицы преобладают над остальными, так как в каждой строке модули этих элементов больше суммы модулей двух остальных элементов, каждый из которых равен единице. При выполнении этого условия решение системы линейных уравнений существует и единственно (см. гл. 4).

Что касается сходимости решения, то здесь имеет место следующее утверждение.

Утверждение. *Если функции $p(x)$ и $f(x)$ дважды непрерывно дифференцируемы, то при $h \to 0$ разностное решение равномерно сходится к точному со скоростью $O(h^2)$.*

Это — достаточное условие сходимости метода конечных разностей для краевой задачи (7.54), (7.55).

Система линейных алгебраических уравнений (7.56) с трехдиагональной матрицей может быть решена методом прогонки (см. гл. 4, § 2, п. 4). При этом условие $p(x) > 0$ гарантирует выполнение условия устойчивости прогонки.

Этот метод на практике используется также и при $p(x) < 0$, хотя успешный результат заранее предвидеть трудно. Для оценки получаемого решения в этом случае необходимо провести расчеты для разных значений шага (не менее трех) и убедиться в том, что полученные значения функции в одних и тех же узлах близки между собой и разность их уменьшается, что говорит о стремлении решения к некоторому пределу при $h \to 0$.

Мы рассмотрели простейший случай линейного уравнения. Значительно труднее решать нелинейные задачи. Рассмотрим краевую задачу для уравнения второго порядка

$$Y''(x) = f(x, Y), \quad 0 \leqslant x \leqslant 1, \tag{7.58}$$
$$Y(0) = A, \quad Y(1) = B.$$

Используя метод конечных разностей, получаем систему разностных нелинейных уравнений

$$y_{i-1} - 2y_i + y_{i+1} = h^2 f(x_i, y_i), \tag{7.59}$$

$$y_0 = A, \quad y_n = B. \tag{7.60}$$

В теории разностных схем доказывается, что разностное решение, определяемое разностными уравнениями (7.59), при $h \to 0$ сходится к точному. Достаточное условие сходимости имеет вид

$$\frac{\partial f}{\partial Y} > 0. \tag{7.61}$$

Система нелинейных алгебраических уравнений (7.59) может быть решена итерационными методами (см. гл. 5, § 3). Для ее решения используют также *метод линеаризации*, т. е. сведение решения нелинейной системы к решению последовательности систем линейных алгебраических уравнений.

Пусть найдено решение системы (7.59) на k-й итерации. Тогда, подставляя известные значения $y_i^{(k)}$ в правые части системы (7.59), получаем

$$y_{i-1}^{(k+1)} - 2y_i^{(k+1)} + y_{i+1}^{(k+1)} = h^2 f\big(x_i, y_i^{(k)}\big).$$

Следовательно, мы пришли к решению системы линейных алгебраических уравнений относительно значений y_i на $(k+1)$-й итерации. Поскольку матрица этой системы трехдиагональна, то для ее решения на каждой итерации может быть использован метод прогонки. Требуется лишь задать некоторые начальные приближения $y_i^{(0)}$ $(i = 1, 2, \ldots, n-1)$; значения y_0, y_n при этом определены граничными условиями (7.60).

Следует отметить, что сходимость данного итерационного процесса довольно медленная. Достаточное условие сходимости имеет вид

$$\frac{1}{8} \max \left| \frac{\partial f}{\partial Y} \right| < 1.$$

Это условие, а также условие (7.61) накладывают ограничения на правую часть $f(x, Y)$ исходного уравнения (7.58).

Упражнения

1. Количество вещества x, участвующего в некоторой химической реакции, определяется уравнением $dx/dt = -x$ (t — время). Найти количество вещества при $t = 10$ с, если в начальный момент оно равно 0.4 моль. Решение провести численным методом, результат сравнить с точным аналитическим решением.

2. Полный магнитный поток Ф катушки, равномерно намотанной на сердечник прямоугольного сечения, определяется уравнением

$$\frac{d\Phi}{dr} = \frac{\mu I n h}{2\pi r}.$$

Определить Ф при следующих данных: $I = 1$ А; $\mu = 0$; размеры катушки: внутренний радиус $R_1 = 4$ см, внешний радиус $R_2 = 6$ см, высота $h = 3$ см, число витков $n = 1500$. Численное решение сравнить с точным.

3. Изменить алгоритм метода Эйлера (см. рис. 7.1) так, чтобы результаты выводились все сразу после полного решения задачи.

4. Исследовать устойчивость задачи Коши для уравнения $y' = ky$, решая это уравнение аналитически и задавая погрешность в определении координат начальной точки.

5. Записать алгоритм решения задачи Коши для системы двух уравнений первого порядка методом Эйлера.

6. Записать алгоритм решения уравнения первого порядка методом Эйлера с пересчетом.

7. Записать алгоритм решения задачи Коши для уравнения второго порядка модифицированным методом Эйлера с автоматическим выбором шага.

8. Показать, что усовершенствованный метод Эйлера имеет второй порядок точности.

9. Получить формулу метода Адамса (7.31).

10. С помощью итерационного метода предиктор-корректор найти решение при $x = 4h$ и $x = 5h$ ($h = 0.1$) для следующей задачи Коши:

$$\frac{dY}{dt} = t + \frac{\sin Y}{3}, \quad Y(0) = 0.3.$$

11. Получить формулу метода Рунге (7.34).

12*. Получить формулу для оценки порядка точности (7.35).

13. Записать алгоритм решения краевой задачи методом стрельбы с использованием метода деления отрезка пополам.

14. Записать алгоритм решения краевой задачи для уравнения $Y'' - p(x)Y = f(x)$ с граничными условиями общего вида.

ГЛАВА 8

УРАВНЕНИЯ С ЧАСТНЫМИ ПРОИЗВОДНЫМИ

§ 1. Элементы теории разностных схем

1. Вводные замечания. В гл. 7 рассматривались обыкновенные дифференциальные уравнения. Их решения зависят лишь от одной переменной: $y = y(x), u = u(t)$ и т. д. Во многих практических задачах искомые функции зависят от нескольких переменных, и описывающие такие задачи уравнения могут содержать частные производные искомых функций. Они называются *уравнениями с частными производными*.

К решению дифференциальных уравнений с частными производными приводят, например, многие задачи механики сплошных сред. Здесь в качестве искомых функций обычно служат плотность, температура, напряжение и др., аргументами которых являются координаты рассматриваемой точки пространства, а также время.

Полная математическая постановка задачи наряду с дифференциальными уравнениями содержит также некоторые дополнительные условия. Если решение ищется в ограниченной области, то задаются условия на ее границе, называемые *граничными* (*краевыми*) *условиями*. Такие задачи называются *краевыми задачами* для уравнений с частными производными.

Если одной из независимых переменных в рассматриваемой задаче является время t, то задаются некоторые условия (например, значения искомых параметров) в начальный момент t_0, называемые *начальными условиями*. Задача, которая состоит в решении уравнения при заданных начальных условиях, называется *задачей Коши* для уравнения с частными производными. При этом задача решается в неограниченном пространстве и граничные условия не задаются. Задачи, при формулировке которых ставятся граничные и начальные условия, называются *нестационарными* (или *смешанными*) *краевыми задачами*. Получающиеся при этом решения меняются с течением времени.

В дальнейшем будем рассматривать лишь *корректно поставленные задачи*, т. е. задачи, решение которых существует и единственно в некотором классе начальных и граничных условий и непрерывно зависит как от этих условий, так и от коэффициентов уравнений. Решение некорректно поставленных задач выходит за рамки данного краткого курса.

Решение простейших задач для уравнений с частными производными в ряде случаев может быть проведено *аналитическими методами*, рассматриваемыми в соответствующих разделах математики. Это относится в основном к некоторым уравнениям первого порядка, а также к уравнениям второго порядка с постоянными коэффициентами. Аналитические

методы полезны не только тем, что дают возможность получать общие решения, которые могут быть использованы многократно. Они имеют также огромное значение для построения численных методов. Проверка разностных схем на известных решениях простейших уравнений позволяет оценить эти схемы, выяснить их сильные и слабые стороны.

Данная глава посвящена численным методам решения задач для уравнений с частными производными. Это основной класс методов, с помощью которых в настоящее время решаются прикладные задачи, моделируемые уравнениями с частными производными. Численные методы требуют наличия компьютеров большой мощности, т. е. обладающих большим объемом памяти и высокой скоростью вычислений.

Среди численных методов широко распространенными являются *разностные методы*. Как и в случае обыкновенных дифференциальных уравнений (см. гл. 7), они основаны на введении некоторой *разностной сетки* в рассматриваемой области. Значения производных, начальные и граничные условия выражаются через значения функций в *узлах сетки*, в результате чего получается система алгебраических уравнений, называемая *разностной схемой*. Решая эту систему уравнений, можно найти в узлах сетки значения *сеточных функций*, которые приближенно считаются равными значениям искомых функций.

Излагаемые в этой главе численные методы применимы к различным типам задач. Мы будем рассматривать лишь достаточно узкий класс задач для уравнений первого и второго порядков, линейных относительно производных. (Напомним, что порядок дифференциального уравнения определяется порядком старшей производной.) В случае двух независимых переменных x, y эти уравнения можно записать в виде

$$a\frac{\partial^2 u}{\partial x^2} + 2b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu = g. \qquad (8.1)$$

Здесь $u = u(x, y)$ — искомая функция. Коэффициенты a, b, c, d, e, f и правая часть g, вообще говоря, могут зависеть от переменных x, y и искомой функции u. В связи с этим уравнение (8.1) может быть: а) с постоянными коэффициентами; б) линейным, если g линейно зависит от u, а коэффициенты зависят только от x, y; в) квазилинейным, если коэффициенты зависят от u; это самый общий вид уравнения (8.1).

Существуют различные виды уравнений в зависимости от соотношения между коэффициентами. Рассмотрим некоторые из них. При $a = b = c = f = 0$, $d \neq 0$, $e \neq 0$ получается уравнение первого порядка вида

$$\frac{\partial u}{\partial x} + p\frac{\partial u}{\partial y} = q,$$

называемое *уравнением переноса*. На практике в этом уравнении одной из переменных может быть время t. Тогда его называют также *эволюционным уравнением*.

Если хотя бы один из коэффициентов a, b, c отличен от нуля, то (8.1) является уравнением второго порядка. В зависимости от знака дискриминанта $D = b^2 - ac$ оно может принадлежать к одному из трех типов: *гиперболическому* ($D > 0$), *параболическому* ($D = 0$) или *эллиптическому* ($D < 0$).

Приведем примеры уравнений с частными производными второго порядка, которые будем в дальнейшем рассматривать:

волновое уравнение (гиперболическое)

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2};$$

уравнение теплопроводности или *диффузии* (параболическое)

$$\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2}, \quad a > 0;$$

уравнение Лапласа (эллиптическое)

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

Если правая часть последнего уравнения отлична от нуля, то оно называется *уравнением Пуассона*.

Приведенные уравнения называются *уравнениями математической физики*. К их решению сводятся многие прикладные задачи. Прежде чем переходить к обсуждению численных методов решения указанных уравнений, рассмотрим основные вопросы построения разностных схем.

2. О построении разностных схем. Как уже отмечалось, построение разностных схем решения уравнений с частными производными основано на введении сетки в рассматриваемом пространстве. Узлы сетки являются расчетными точками.

Пример простейшей прямоугольной области $G(x, y)$ с границей Γ в двумерном случае показан на рис. 8.1. Стороны прямоугольника $a \leqslant x \leqslant b$, $c \leqslant y \leqslant d$ делятся на элементарные отрезки точками $x_i = a + ih_1$ ($i = 0, 1, \ldots, I$) и $y_j = c + jh_2$ ($j = 0, 1, \ldots, J$). Через эти точки проводятся

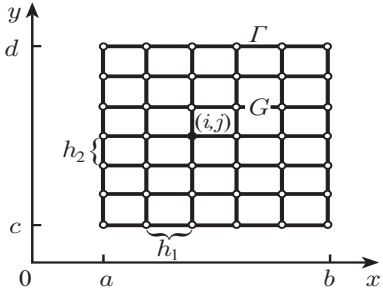


Рис. 8.1. Прямоугольная сетка

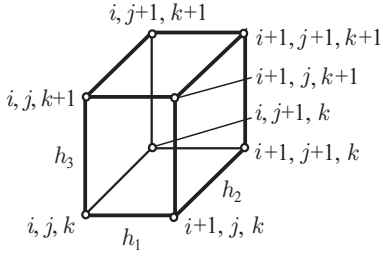


Рис. 8.2. Элемент сетки

два семейства координатных прямых $x = \text{const}$ и $y = \text{const}$, образующих сетку с прямоугольными ячейками. Любой ее узел, номер которого (i, j), определяется координатами (x_i, y_j). Поскольку все ячейки показанной на рис. 8.1 сетки одинаковы, такую сетку называют *равномерной*.

Аналогично вводятся сетки для многомерных областей, содержащих более двух измерений. На рис. 8.2 показан элемент сетки в виде прямоугольного параллелепипеда для трехмерной области.

Прямоугольные сетки наиболее удобны при организации вычислительного алгоритма. Вместе с тем некоторые схемы используют сетки с ячейками более сложной формы: треугольными, четырехугольными (не прямоугольными), шестиугольными и т. д.

Узлы сетки, лежащие на границе Γ области G, называются *граничными узлами*. Все остальные узлы — *внутренними*. Поскольку начальные и граничные условия при постановке задач формулируются на границе расчетной области, то их можно считать заданными в граничных узлах сетки. Иногда граничные точки области не являются узлами сетки, что имеет место для областей сложной формы. Тогда либо вводят дополнительные узлы на пересечении координатных линий с границей, либо границу приближенно заменяют ломаной, проходящей через близкие к границе узлы. На эту ломаную переносятся граничные условия.

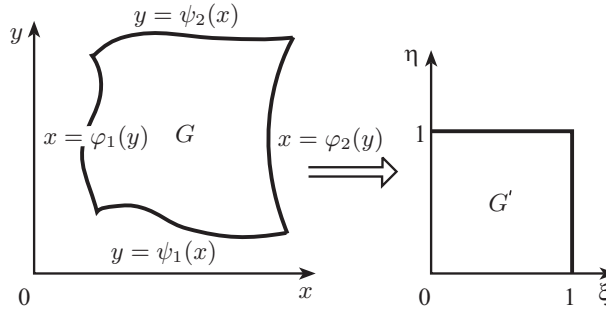


Рис. 8.3. Преобразование расчетной области

В ряде случаев сложные криволинейные области с помощью перехода к новым независимым переменным удается свести к простейшему виду. Например, четырехугольную область G, изображенную на рис. 8.3, можно привести к единичному квадрату G' путем введения новых переменных ξ, η вместо x, y с помощью соотношений

$$\xi = \frac{x - \varphi_1(y)}{\varphi_2(y) - \varphi_1(y)}, \quad 0 \leqslant \xi \leqslant 1,$$

$$\eta = \frac{y - \psi_1(x)}{\psi_2(x) - \psi_1(x)}, \quad 0 \leqslant \eta \leqslant 1.$$

К новым переменным нужно преобразовать уравнения, а также начальные

и граничные условия. В области G' можно ввести прямоугольную сетку, при этом в области G ей будет соответствовать сетка с неравномерно расположенными узлами и криволинейными ячейками.

В дальнейшем при построении разностных схем мы для простоты будем использовать прямоугольные сетки (или с ячейками в виде прямоугольных параллелепипедов в трехмерном случае), а уравнения будем записывать в декартовых координатах (x, y, z). На практике приходится решать задачи в различных криволинейных системах координат: полярной, цилиндрической, сферической и др. Например, если расчетную область удобно задать в полярных координатах (r, φ), то в ней сетка вводится с *шагами* Δr и $\Delta\varphi$ соответственно по радиус-вектору и полярному углу.

Иногда и в простой расчетной области вводят *неравномерную сетку*. В частности, в ряде случаев необходимо проводить сгущение узлов для более точного расчета в некоторых частях рассматриваемой области. При этом области сгущения узлов либо известны заранее, либо определяются в процессе решения задачи (например, в зависимости от градиентов искомых функций). В последнем случае получающиеся сетки называют *адаптивными*.

Для построения разностной схемы, как и в случае обыкновенных дифференциальных уравнений, частные производные в уравнении заменяются конечно-разностными соотношениями по некоторому шаблону (см. гл. 3, § 1). При этом точные значения искомой функции заменяются значениями сеточной функции в узлах разностной сетки.

В качестве примера построим некоторые разностные схемы для решения уравнения теплопроводности при заданных начальных и граничных условиях. Запишем смешанную краевую задачу в виде

$$\frac{\partial U}{\partial t} = a\,\frac{\partial^2 U}{\partial x^2}, \quad 0 \leqslant x \leqslant 1, \quad t > 0, \quad a > 0,$$
$$U(x,0) = \varphi(x), \quad U(0,t) = \psi_1(t), \quad U(1,t) = \psi_2(t), \tag{8.2}$$

где $\varphi(x)$ — начальное распределение температуры U (при $t = 0$); $\psi_1(t)$, $\psi_2(t)$ — распределение температуры на концах рассматриваемого отрезка ($x = 0,\ 1$) в любой момент времени t. Заметим, что начальные и граничные условия должны быть согласованы, т. е. $U(0,0) = \varphi(0) = \psi_1(0)$, $U(1,0) = \varphi(1) = \psi_2(0)$.

Введем равномерную прямоугольную сетку с помощью координатных линий $x_i = ih$ ($i = 0, 1, \ldots, I$), $t_j = j\tau$ ($j = 0, 1, \ldots$); h и τ — соответственно шаги сетки по направлениям x и t. Значения функции в узлах сетки обозначим $U_i^j = U(x_i, t_j)$. Эти значения заменим соответствующими значениями сеточной функции u_i^j, которые удовлетворяют уравнениям, образующим разностную схему [1)].

[1)] Часто верхний индекс заключают в скобки, чтобы не путать его с показателем степени. Здесь и далее скобки для краткости опущены.

Заменяя в исходном уравнении (8.2), частные производные искомой функции с помощью отношений конечных разностей, получаем разностную схему

$$\frac{u_i^{j+1}-u_i^j}{\tau}=a\,\frac{u_{i+1}^j-2u_i^j+u_{i-1}^j}{h^2}\,,\qquad \begin{matrix} i=1,2,\ldots,I-1, \quad j=1,2,\ldots \end{matrix} \tag{8.3}$$

В записи этой схемы для каждого узла использован шаблон, изображенный на рис. 8.4, *а*.

Для одного и того же уравнения можно построить различные разностные схемы. В частности, если воспользоваться шаблоном, изображенным

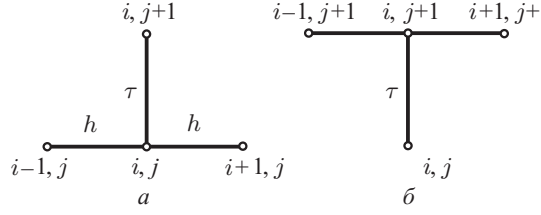


Рис. 8.4. Шаблоны

на рис. 8.4, *б*, т. е. аппроксимировать производную $\partial^2U/\partial x^2$ при $t=t_{j+1}$:

$$\frac{\partial^2U}{\partial x^2}\approx\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{h^2}\,.$$

то вместо (8.3) получим разностную схему

$$\frac{u_i^{j+1}-u_i^j}{\tau}=a\,\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{h^2}\,, \tag{8.4}$$

И в том и другом случае получается система алгебраических уравнений для определения значений сеточной функции во внутренних узлах. Значения в граничных узлах находятся из граничных условий

$$u_0^j=\psi_1(t_j),\quad u_I^j=\psi_2(t_j). \tag{8.5}$$

Совокупность узлов при $t=\mathrm{const}$, т. е. при фиксированном значение j, называется *слоем* (или, поскольку переменная t соответствует времени, *временны́м слоем*). Схема (8.3) позволяет последовательно находить значения u_i^{j+1} $(i=1,2,\ldots,I-1)$ на $(j+1)$-м слое через соответствующие значения u_i^j на j-м слое. Такие схемы называются *явными*.

Для начала счета по схеме (8.3) при $j=1$ необходимо знать решение на начальном слое при $j=0$. Оно определяется начальным условием (8.2),

которое запишется в виде

$$u_i^0 = \varphi(x_i), \quad i = 1, 2, \ldots, I-1. \tag{8.6}$$

В отличие от явной схемы каждое разностное уравнение (8.4) содержит на каждом новом слое три неизвестных значения: u_{i-1}^{j+1}, u_i^{j+1}, u_{i+1}^{j+1}, поэтому нельзя сразу определить эти значения через известное решение на предыдущем слое. Такие схемы называются *неявными*. При этом разностная схема (8.4) состоит из линейных трехточечных уравнений, т. е. каждое уравнение содержит неизвестную функций в трех точках данного слоя. Такие системы линейных уравнений с трехдиагональной матрицей могут быть решены методом прогонки (см. гл. 4, § 2, п. 4), в результате чего будут найдены значения сеточной функции в узлах.

Заметим, что в рассмотренном примере мы получаем *двухслойные схемы*, когда в каждое разностное уравнение входят значения функции из двух слоев: нижнего, на котором решение уже найдено, и верхнего, в узлах которого решение ищется.

С помощью рассматриваемого способа построения разностных схем, когда входящие в уравнение отдельные частные производные заменяются конечно-разностными соотношениями для сеточной функции (или сеточными выражениями), могут быть созданы многослойные схемы, а также схемы высоких порядков точности.

Несмотря на то что этот способ получения разностных уравнений наиболее прост и поэтому широко используется при разработке численных методов, существуют также другие способы построения разностных схем. Изложение этих вопросов читатель может найти в более полных работах по численным методам и теории разностных схем, список которых приведен в конце книги.

3. Сходимость. Аппроксимация. Устойчивость. Эти основные понятия теории разностных схем уже обсуждались при построении численных методов для решения обыкновенных дифференциальных уравнений. При переходе к уравнениям с частными производными качественно меняется характер рассматриваемых задач, поэтому необходимо снова рассмотреть эти понятия. Разумеется, мы не имеем здесь возможности изложить теорию разностных схем, но попытаемся привести самые необходимые сведения.

Исходную *дифференциальную задачу*, состоящую в решении уравнения с частными производными при заданных начальных и граничных условиях, запишем в операторном виде:

$$LU(x,t) = F(x,t), \quad (x,t) \in \overline{G}. \tag{8.7}$$

Заметим, что это операторное уравнение включает не только исходное уравнение с частными производными, но и дополнительные (начальные и граничные) условия. Функция $F(x,t)$ описывает правые части уравнения, а также начальные и граничные условия. Область $\overline{G}$ включает расчетную область G и границу Γ.

Дифференциальную задачу (8.7) заменяем разностной задачей относительно сеточной функции u_h определенной в узлах сетки $\overline{g}_h$. Для простоты будем считать, что сетка зависит от одного параметра h, а шаг по времени τ выражается через h: $\tau = rh$, где $r = \text{const}$. Разностную задачу можно также записать в операторном виде:

$$L_h u_h = f_h, \quad (x,t) \in \overline{g}_h. \tag{8.8}$$

Значения сеточной функции u_i^j в узлах сетки $(x_i, t_j) \in \overline{g}_h$ приближенно заменяют значения искомой функции $U_i^j = U(x_i, t_j)$ в тех же узлах с погрешностями

$$\delta u_i^j = U_i^j - u_i^j. \tag{8.9}$$

Введем некоторое характерное значение этих погрешностей, например их максимальное по модулю значение на сетке

$$\delta u = \max_{i,j} \left|\delta u_i^j\right|,$$

Разностная схема (8.8) называется *сходящейся*, если при сгущении узлов сетки это значение погрешности стремится к нулю, т. е. если

$$\lim_{h \to 0} \delta u = 0.$$

Если при этом $\delta u \leqslant Mh^k$, где $M = \text{const} > 0$, то разностная схема имеет *k-й порядок точности*. Говорят также, что она *сходится со скоростью* $O(h^k)$.

Можно ввести понятие порядка точности и для случая независимых параметров сетки h, τ. В частности, при выполнении условия $\delta u \leqslant M(h^p + \tau^q)$ разностная схема сходится со скоростью $O(h^p + \tau^q)$ и имеет p-й порядок точности по h и q-й порядок по τ.

Определим сеточную функцию погрешности δ_h как разность между решением дифференциальной задачи, рассматриваемом в узлах сетки, и разностным решением: $\delta_h = U_h - u_h$. При этом значение δ_h в узле с номером (i, j) определяется соотношением (8.9). Выразим u_h через U_h и δ_h и подставим в уравнение (8.8). Имеем

$$\begin{aligned} u_h = U_h - \delta_h, \quad & L_h U_h - L_h \delta_h = f_h, \\ L_h \delta_h = R_h, \quad & R_h = L_h U_h - f_h. \end{aligned} \tag{8.10}$$

Величина R_h называется *невязкой* (*погрешностью аппроксимации*) разностной схемы. Она равна разности между левой и правой частями (8.8) при подстановке в это уравнение решения дифференциальной задачи (8.7).

Введем некоторую характерную величину невязки R, например

$$R = \max_{(x,t) \in \overline{g}_h} |R_h|.$$

Тогда при $R = O(h^k)$ аппроксимация имеет k-й порядок относительно h. Если значения h и τ независимы, то при $R = O(h^p + \tau^q)$ порядок аппроксимации разностной схемы p-й по пространству и q-й по времени.

Разностная схема (8.8) *аппроксимирует* исходную дифференциальную задачу (8.7), если при измельчении сетки невязка стремится к нулю, т. е. если

$$\lim_{\substack{h\to 0\\ \tau\to 0}} R = 0.$$

Аппроксимация такого типа, т. е. когда невязка стремится к нулю при стремлении к нулю h и τ по любому закону без каких-либо условий, называется *безусловной* или *абсолютной аппроксимацией*. В случае *условной аппроксимации* накладываются некоторые условия на размеры шагов по пространству и времени. Например, если $R = O(h + \tau + \tau/h^2)$, то $R \to 0$ при $h \to 0$, $\tau \to 0$ и $\tau/h^2 \to 0$, т. е. разностная задача аппроксимирует исходную при условии, что τ стремится к нулю быстрее, чем h^2. Так, при $t = h^2$ аппроксимация в данном примере отсутствует.

Разностная схема (8.8) называется *устойчивой*, если ее решение непрерывно зависит от входных данных, т. е. малому изменению входных данных соответствует малое изменение решения. Устойчивость характеризует чувствительность разностной схемы к различного рода погрешностям. Она является внутренним свойством разностной задачи, и это свойство не связывается непосредственно с исходной дифференциальной задачей (в отличие от сходимости и аппроксимации).

По аналогии с аппроксимацией *устойчивость* бывает *условной* и *безусловной* в зависимости от того, накладываются или нет ограничения на соотношения между шагами по разным переменным.

В теории разностных схем рассматриваются разные способы исследования аппроксимации исходной дифференциальной задачи разностной и проверки устойчивости разностных схем. Здесь мы лишь отметим, что эти исследования значительно проще, чем доказательство сходимости разностного решения к точному. Поэтому пользуются следующим утверждением.

Т е о р е м а. *Если решение исходной дифференциальной задачи* (8.7) *существует, а разностная схема* (8.8) *устойчива и аппроксимирует задачу* (8.7) *на данном решении с порядком k, то разностное решение сходится к точному со скоростью $O(h^{(k)})$.*

Короче говоря, из аппроксимации и устойчивости следует сходимость. Поэтому, доказав аппроксимацию и устойчивость разностной схемы, можем быть уверены в ее сходимости.

Проиллюстрируем исследование разностных схем на примере рассмотренных выше двух схем для уравнения теплопроводности — явной схемы (8.3) и неявной схемы (8.4). Будем считать, что решение $U(x,t)$ дифференциальной задачи (8.2) существует, а частные производные $\partial^2 U/\partial t^2$ и $\partial^4 U/\partial x^4$ непрерывны и ограничены в расчетной области. Тогда в соответствии с формулами численного дифференцирования для каждого узла (x_i, t_j) $(i = 1, 2, \ldots, I-1,\ j = 1, 2, \ldots)$ можно написать следующие соотношения:

$$\frac{U_i^{j+1}-U_i^j}{\tau}=\frac{\partial U(x_i,t_j)}{\partial t}+O(\tau),$$
$$\frac{U_{i+1}^j-2U_i^j+U_{i-1}^j}{h^2}=\frac{\partial^2 U(x_i,t_j)}{\partial x^2}+O(h^2). \tag{8.11}$$

Найдем погрешность аппроксимации R_i^j исходного уравнения (8.2) с помощью разностной схемы (8.3) для произвольного узла сетки(x_i,t_j):

$$R_i^j=\frac{U_i^{j+1}-U_i^j}{\tau}-a\,\frac{U_{i+1}^j-2U_i^j+U_{i-1}^j}{h^2}.$$

Подставим в это равенство соотношения (8.11). При этом заметим, что поскольку $U(x,t)$ является точным решением уравнения (8.2), то

$$\frac{\partial U(x_i,t_j)}{\partial t}-a\,\frac{\partial^2 U(x_i,t_j)}{\partial x^2}=0. \tag{8.12}$$

Следовательно, максимальное значение невязки с учетом (8.11), (8.12) имеет порядок

$$R=\max_{i,j}\left|R_i^j\right|=O(h^2)+O(\tau)=O(h^2+\tau).$$

Аналогичную оценку невязки можно получить и для разностной схемы (8.4).

Таким образом, разностные схемы (8.3) и (8.4) аппроксимируют исходное дифференциальное уравнение (8.2) со вторым порядком по h и с первым порядком по τ. Начальное и граничные условия задачи (8.2) аппроксимируются на границах точно, поскольку здесь значения сеточной функции равны значениям решения: $u_i^j=U(x_i,t_j)$, где $(x_i,t_j)\in\Gamma$, Γ — граница расчетной области ($t=0$, $x=0$, $x=1$).

Исследуем теперь устойчивость данных разностных схем. Начнем с явной схемы (8.3) при граничных условиях (8.5) и начальном условии (8.6). Найдем из (8.3) значение u_i^{j+1} сеточной функции на верхнем слое:

$$u_i^{j+1}=\lambda u_{i-1}^j+(1-2\lambda)u_i^j+\lambda u_{i+1}^j,\quad \lambda=a\tau/h^2,$$
$$i=1,2,\ldots,I-1. \tag{8.13}$$

Допустим, что имеет место ограничение в виде неравенства

$$\lambda\leqslant 1/2. \tag{8.14}$$

Тогда $\lambda+|1-2\lambda|+\lambda=\lambda+1-2\lambda+\lambda=1$. Эти соотношения используем для оценки сеточного решения (8.13):

$$\max_{1\leqslant i\leqslant I-1}\left|u_i^{j+1}\right|=\max_{1\leqslant i\leqslant I-1}\left|\lambda u_{i-1}^j+(1-2\lambda)u_i^j+\lambda u_{i+1}^j\right|\leqslant$$
$$\leqslant(\lambda+|1-2\lambda|+\lambda)\max_{0\leqslant i\leqslant I}\left|u_i^j\right|=\max_{0\leqslant i\leqslant I}\left|u_i^j\right|. \tag{8.15}$$

Введем теперь обозначение для наибольшего по модулю значения сеточной функции на j-м слое

$$u_*^j = \max_{0\leqslant i\leqslant I} \left|u_i^j\right|$$

и с учетом граничных условий (8.5) запишем неравенство (8.15) для значений решения на всем $(j+1)$-м слое, включая границы:

$$u_*^{j+1} \leqslant \max\bigl(u_*^j, |\psi_1(t_{j+1})|, |\psi_2(t_{j+1})|\bigr). \tag{8.16}$$

Отсюда при $j=0$ получаем

$$u_*^1 \leqslant \max\bigl(u_*^0, |\psi_1(t_1)|, |\psi_2(t_1)|\bigr). \tag{8.17}$$

Из (8.5), (8.6) следует, что

$$u_*^0 = \max\bigl(\varphi_*, |\psi_1(t_0)|, |\psi_2(t_0)|\bigr), \quad \varphi_* = \max_{1\leqslant i\leqslant, I-1} |\varphi(x_i)|, \quad t_0 = 0,$$

поэтому неравенство (8.17) можно записать в виде

$$u_*^1 \leqslant \max\bigl(\varphi_*, \psi_*^1\bigr), \quad \psi_*^1 = \max_{j=0,1} |\psi_{1,2}(t_j)|. \tag{8.18}$$

При $j=1$ из (8.16), (8.18) получаем

$$u_*^2 \leqslant \max\bigl(u_*^1, |\psi_1(t_2)|, |\psi_2(t_2)|\bigr) \leqslant \max\bigl(\varphi_*, \psi_*^2\bigr), \quad \psi_*^2 = \max_{j=0,1,2} |\psi_{1,2}(t_j)|.$$

Аналогично, для некоторого $j=J$ имеем

$$\begin{aligned} u_*^{J+1} &\leqslant \max(\varphi_*, \psi_*^{J+1}), \\ \psi_*^{J+1} &= \max_{0\leqslant j\leqslant J+1} |\psi_{1,2}(t_j)|. \end{aligned} \tag{8.19}$$

Таким образом, значения сеточного решение на $(J+1)$-м слое не превосходят по модулю известных значений сеточного решения на нулевом слое ($j=0$) и на границах $i=0$, $i=I$ (по $(J+1)$-й слой включительно).

Неравенство (8.19) означает устойчивость разностной схемы (8.3). Покажем это. Разностная схема была выше названа устойчивой, если малому изменению входных данных соответствует малое изменение решения. Рассмотрим разностную задачу, входные данные которой, например, начальное условие, подверглись малому изменению $\tilde\varphi(x_i)$:

$$\begin{gathered} \frac{v_i^{j+1} - v_i^j}{\tau} = a\,\frac{v_{i+1}^j - 2v_i^j + v_{i-1}^j}{h^2}, \\ v_i^0 = \varphi(x_i) + \tilde\varphi(x_i), \quad v_0^j = \psi_1(t_j), \quad v_I^j = \psi_2(t_j). \end{gathered} \tag{8.20}$$

Решением этой задачи будет сеточная функция

$$v_i^j = u_i^j + \tilde u_i^j, \tag{8.21}$$

где u_i^j — решение исходной разностной задачи (8.3), (8.5), (8.6), а $\tilde{u}_i^j$ — некоторая поправка к решению. Подставим (8.21) в (8.20):

$$\frac{u_i^{j+1}-u_i^j}{\tau}+\frac{\tilde{u}_i^{j+1}-\tilde{u}_i^j}{\tau}=a\,\frac{u_{i+1}^j-2u_i^j+u_{i-1}^j}{h^2}+a\,\frac{\tilde{u}_{i+1}^j-2\tilde{u}_i^j+\tilde{u}_{i-1}^j}{h^2},$$
$$u_i^0+\tilde{u}_i^0=\varphi(x_i)+\tilde{\varphi}(x_i),\quad u_0^j+\tilde{u}_0^j=\psi_1(t_j),\quad u_I^j+\tilde{u}_I^j=\psi_2(t_j).$$

Отсюда с учетом (8.3), (8.5), (8.6) получаем разностную задачу относительно поправки $\tilde{u}_i^j$

$$\frac{\tilde{u}_i^{j+1}-\tilde{u}_i^j}{\tau}=a\,\frac{\tilde{u}_{i+1}^j-2\tilde{u}_i^j+\tilde{u}_{i-1}^j}{h^2},$$
$$\tilde{u}_i^0=\tilde{\varphi}(x_i),\quad \tilde{u}_0^j=0,\quad \tilde{u}_I^j=0.$$

Эта задача совпадает с исходной, но при других начальных и граничных условиях. К ее решению $\tilde{u}_i^j$ применимо неравенство (8.19), которое в данном случае имеет вид

$$\tilde{u}_*^{J+1}\leqslant\tilde{\varphi}_*$$

и означает малость поправки к решению при малом изменении начального условия. Таким образом, схема (8.3) устойчива при выполении условия (8.14). Можно показать, что при нарушении этого условия схема (8.3) будет неустойчивой, т. е. явная схема (8.3) условно устойчива. Из аппроксимации и устойчивости следует ее сходимость со скоростью $O(h^2+\tau)$.

Исследуем теперь устойчивость неявной разностной схемы (8.4). Запишем, используя (8.4), (8.5), систему уравнений для нахождения неизвестных значений сеточной функции на верхнем слое:

$$\begin{gathered}\lambda u_{i-1}^{j+1}-(1+2\lambda)u_i^{j+1}+\lambda u_{i+1}^{j+1}=-u_i^j,\quad i=1,2,\ldots,I-1,\\ u_0^{j+1}=\psi_1(t_{j+1}),\quad u_I^{j+1}=\psi_2(t_{j+1}).\end{gathered}\tag{8.22}$$

Эта система может быть решена методом прогонки. Безусловная устойчивость неявной схемы (8.4) обеспечивается выполнением условий устойчивости метода прогонки для системы (8.22).

Оценки устойчивости и сходимости разностных схем можно провести путем расчетов с измельчением сетки ($h\to 0$, $\tau\to 0$). Однако это приводит к существенному увеличению объема вычислений и возрастанию суммарных погрешностей.

Многолетняя практика использования численных методов для решения инженерных задач на компьютерах показывает, что применение той или иной разностной схемы, даже если она исследована теоретически, требует ее тщательной апробации при решении конкретной задачи. Для этого проводятся методические вычислительные эксперименты, состоящие в расчетах с разными значениями шагов при разных исходных данных. Полезно также отладить методику с помощью тестовых задач, для которых либо

удается получить аналитическое решение, либо имеется численное решение, найденное другим численным методом.

§ 2. Уравнения первого порядка

1. Линейное уравнение переноса. При классификации уравнений с частными производными (8.1) отмечалось, что уравнения первого порядка называются также уравнениями переноса. Это объясняется тем, что такие уравнения описывают процессы переноса частиц в средах, распространения возмущений и т. п.

В общем случае уравнения переноса могут иметь значительно более сложный вид (например, интегродифференциальное уравнение Больцмана в кинетической теории газов). Однако здесь мы ограничимся линейным уравнением с частными производными первого порядка. Его решение представляет интерес не только с практической точки зрения; в еще большей степени это уравнение полезно при разработке и исследовании разностных схем.

Будем считать, что искомая функция U зависит от времени t и одной пространственной переменной x. Тогда линейное уравнение переноса может быть записано в виде

$$\frac{\partial U}{\partial t} + a\,\frac{\partial U}{\partial x} = F(x,t). \tag{8.23}$$

Здесь a — скорость переноса, которую будем считать постоянной и положительной. Это соответствует переносу (распространению возмущений) слева направо в положительном направлении оси x. Правая часть $F(x,t)$ характеризует наличие поглощения (или, наоборот, источников) энергии, частиц и т. д. в зависимости от того, какой физический процесс описывается уравнением переноса.

Характеристики уравнения (8.23) определяются соотношениями $x - at = C = \text{const}$. При постоянном a они являются прямыми линиями, которые в данном случае ($a > 0$) наклонены вправо (рис. 8.5).

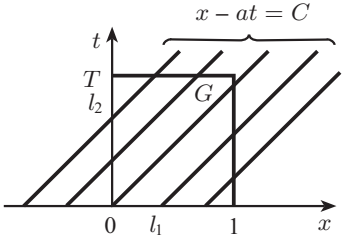


Рис. 8.5. Область решения

Расчетная область при решении уравнения (8.23) может быть как бесконечной, так и ограниченной. В первом случае, задавая начальное условие при $t = 0$:

$$U(x,0) = \Phi(x), \tag{8.24}$$

получаем задачу Коши для полуплоскости ($t \geqslant 0$, $-\infty < x < +\infty$). На практике обычно приходится решать уравнение переноса в некоторой ограниченной области (например, в прямоугольнике $0 \leqslant x \leqslant 1$, $0 \leqslant t \leqslant T$; см. рис. 8.5). Начальное условие (8.24) в этом случае

задается на отрезке l_1; граничное условие нужно задать при $x = 0$, т. е. на отрезке l_2, поскольку при $a > 0$ возмущения распространяются вправо. Это условие запишем в виде

$$U(0, t) = \Psi(t). \tag{8.25}$$

Таким образом, задача состоит в решении уравнения (8.23) с начальным и граничным условиями (8.24) и (8.25) в ограниченной области G: $0 \leqslant x \leqslant 1$, $0 \leqslant t \leqslant T$.

Убедиться в том, что данная задача поставлена правильно (корректно) можно, проанализировав решение уравнения (8.23), которое при $F(x, t) = 0$ имеет вид

$$U(x, t) = H(x - at), \tag{8.26}$$

где H — произвольная дифференцируемая функция. В этом легко убедиться, подставляя (8.26) в уравнение (8.23). Решение (8.26) называется *бегущей волной* (со скоростью a). Это решение постоянно вдоль каждой характеристики: при $x - at = C$ искомая функция $U = H(x - at) = H(C)$ постоянна. Таким образом, начальные и граничные условия переносятся вдоль характеристик, поэтому они должны задаваться на отрезках l_1 и l_2 расчетной области G (см. рис. 8.5).

Можно также построить аналитическое решение задачи Коши для неоднородного уравнения (8.23). Заметим лишь, что решение этой задачи меняется вдоль характеристики, а не является постоянным.

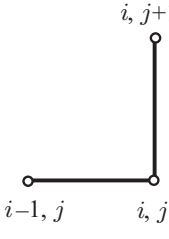


Рис. 8.6. Правый нижний уголок

Рассмотрим разностные схемы для решения задачи (8.23) – (8.25). Построим в области G равномерную прямоугольную сетку с помощью прямых $x_i = ih$ ($i = 0, 1, \ldots, I$) и $t_j = j\tau$ ($j = 0, 1, \ldots, J$). Вместо функций $U(x, t)$, $F(x, t)$, $\Phi(x)$ и $\Psi(t)$ будем рассматривать сеточные функции, значения которых в узлах (x_i, t_j) соответственно равны u_i^j, f_i^j, φ_i и ψ^j. Для построения разностной схемы необходимо выбрать шаблон. Примем его в виде *правого нижнего уголка* (рис. 8.6). При этом входящие в уравнение (8.23) производные аппроксимируются конечно-разностными соотношениями с использованием односторонних разностей:

$$\frac{u_i^{j+1} - u_i^j}{\tau} + a\,\frac{u_i^j - u_{i-1}^j}{h} = f_i^j. \tag{8.27}$$

Решая это разностное уравнение относительно единственного неизвестного значения u_i^{j+1} на $(j + 1)$-м слое, получаем следующую разностную схему:

$$\begin{gathered} u_i^{j+1} = \lambda u_{i-1}^j + (1 - \lambda) u_i^j + \tau f_i^j, \\ \lambda = a\tau/h, \quad i = 1, 2, \ldots, I, \quad j = 0, 1, \ldots, J - 1. \end{gathered} \tag{8.28}$$

Полученная схема явная, поскольку значения сеточной функции в каждом узле верхнего слоя $t = t_{j+1}$ выражаются явно с помощью соотношений (8.28) через ранее найденные ее значения на предыдущем слое.

Для начала счета по схеме (8.28), т. е. для вычисления сеточной функции на первом слое, необходимы ее значения на слое $j = 0$. Они определяются начальным условием (8.24), которое записываем для сеточной функции:

$$u_i^0 = \varphi_i, \quad i = 0, 1, \ldots, I. \tag{8.29}$$

Граничное условие (8.25) также записывается в сеточном виде:

$$u_0^j = \psi^j, \quad j = 1, 2, \ldots, J. \tag{8.30}$$

Таким образом, решение исходной дифференциальной задачи (8.23) – (8.25) сводится к решению разностной задачи (8.28) – (8.30). Найденные значения сеточной функции u_i^j принимаются в качестве значений искомой функции и в узлах сетки.

Алгоритм решения исходной задачи (8.23) – (8.25) с применением рассмотренной разностной схемы достаточно прост. На рис. 8.7 представлена его структурограмма. В соответствии с этим алгоритмом в памяти компьютера хранится весь двумерный массив u_i^j, и он целиком выводится на печать по окончании счета. С целью экономии памяти (и если эти результаты не понадобятся для дальнейшей обработки) можно воспользоваться тем, что схема двухслойная, и хранить лишь значения сеточной функции на двух соседних слоях u_i^j, u_i^{j+1}. Рекомендуем читателю соответственным образом модифицировать представленный алгоритм и построить новую структурограмму.

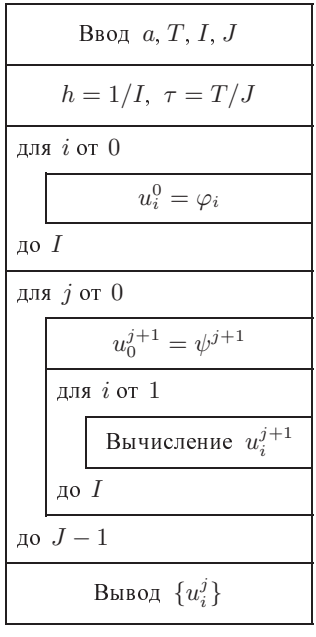


Рис. 8.7. Алгоритм решения линейного уравнения переноса

Укажем теперь некоторые свойства данной разностной схемы. Она аппроксимирует исходную задачу с первым порядком, т. е. невязка имеет порядок $O(h + \tau)$. Схема условно устойчива; условие устойчивости имеет вид

$$0 < \tau \leqslant h/a. \tag{8.31}$$

Эти свойства схемы установлены в предположении, что решение $U(x, t)$, начальное и граничное значения $\Phi(x)$ и $\Psi(t)$ дважды непрерывно дифференцируемы, а правая часть $F(x, t)$ имеет непрерывные первые производные.

Поскольку схема (8.28) устойчива и аппроксимирует исходную задачу, то в соответствии с приведенной в § 1 теоремой сеточное решение сходится к точному с первым порядком при $h \to 0$, $\tau \to 0$. Отметим, что при $a < 0$ условие (8.31) не выполняется, и схема (8.28) не сходится.

Можно построить сходящуюся схему и для случая $a < 0$. В качестве шаблона для посторения разностной схемы для уравнения (8.23) примем *левый нижний уголок* (рис. 8.8). Разностное уравнение в этом случае примет вид

$$\frac{u_i^{j+1} - u_i^j}{\tau} + a\,\frac{u_{i+1}^j - u_i^j}{h} = f_i^j. \tag{8.32}$$

Эта схема является условно устойчивой (следовательно, сходящейся) при $a < 0$, если выполнено соотношение

$$\tau \leqslant -h/a,$$

которое аналогично условию (8.31). При $a > 0$ эта схема не сходится.

Граничное условие для уравнения переноса (8.23) при $a < 0$ задается при $x = 1$, поскольку возмущения в данном случае распространяются влево:

$$U(1, t) = \Psi(t).$$

Соответствующее граничное условие для разностного уравнения (8.32) имеет вид

$$u_I^j = \psi^j, \quad j = 1, 2, \ldots, J. \tag{8.33}$$

Сравним две рассмотренные схемы, построенные на шаблонах типа правый и левый уголок. Обе используют для аппроксимации производной $\partial U/\partial x$ в узле (x_i, t_j) значение функции в другом узле, который расположен на том же слое и отстоит от узла (x_i, t_j) в направлении, противоположном направлению распространения возмущений (направлению *потока*). Например, правый уголок содержит узел (x_{i-1}, t_j), расположенный левее узла (x_i, t_j), в то время как возмущения распространяются вправо ($a > 0$). Такая аппроксимация называется *противопотоковой* и широко используется при численном решении уравнений переноса.

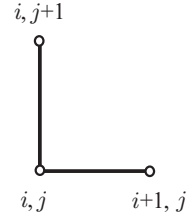


Рис. 8.8. Левый нижний уголок

При построении явной разностной схемы (8.28) производная $\partial U/\partial x$ аппроксимировалась с помощью значений сеточной функции на j-м слое; в результате получилось разностное уравнение (8.27), в котором использовано значение сеточной функции u_i^{j+1} лишь в одном узле верхнего слоя. Если производную $\partial U/\partial x$ аппроксимировать на $(j+1)$-м слое (шаблон изображен на рис. 8.9), то получится неявная схема. Разностное уравнение примет вид

$$\frac{u_i^{j+1} - u_i^j}{\tau} + a\,\frac{u_i^{j+1} - u_{i-1}^{j+1}}{h} = f_i^j. \tag{8.34}$$

Разрешая это уравнение относительно u_i^{j+1}, приходим к следующей разностной схеме:

$$u_i^{j+1} = \frac{u_i^j + \lambda u_{i-1}^{j+1} + \tau f_i^j}{1+\lambda}, \quad \lambda = \frac{a\tau}{h}. \tag{8.35}$$

Это двухслойная трехточечная схема первого порядка точности. Она безусловно устойчива (при $a > 0$). Хотя формально данная разностная схема строилась как неявная, практическая организация счета по ней проводится так же, как и для явных схем.

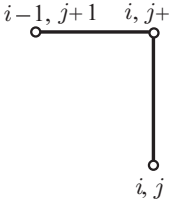


Рис. 8.9. Правый верхний уголок

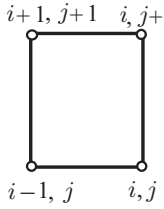


Рис. 8.10. Прямоугольник

Действительно, в правую часть уравнения (8.35) входит значение u_{i-1}^{j+1} на $(j+1)$-м слое, которое при вычислении u_i^{j+1} уже найдено. При расчете u_1^{j+1} значение u_0^{j+1} берется из граничного условия (8.30). По объему вычислений и логике программы (см. рис. 8.7) схема (8.35) аналогична схеме (8.28), однако безусловная устойчивость делает ее более удобной, поскольку исключается ограничение на величину шага.

Схему (8.28) можно применять для решения задачи Коши в неограниченной области, поскольку граничное условие (8.30) в этой схеме можно не использовать.

Рассмотрим еще одну разностную схему, которую построим на симметричном прямоугольном шаблоне (рис. 8.10). Производная по t здесь аппроксимируется в виде полусуммы отношений односторонних конечных разностей в $(i-1)$-м и i-м узлах, а производная по x — в виде полусуммы конечно-разностных соотношений на j-м и $(j+1)$-м слоях. Правая часть вычисляется в центре ячейки, хотя возможны и другие способы ее вычисления (например, в виде некоторой комбинации ее значений в узлах). В результате указанных аппроксимаций получим разностное уравнение в виде

$$\frac{1}{2}\left(\frac{u_{i-1}^{j+1} - u_{i-1}^j}{\tau} + \frac{u_i^{j+1} - u_i^j}{\tau}\right) + \frac{a}{2}\left(\frac{u_i^j - u_{i-1}^j}{h} + \frac{u_i^{j+1} - u_{i-1}^{j+1}}{h}\right) = \bar{f}_i^j, \tag{8.36}$$

$$\bar{f}_i^j = f(x_i + h/2,\, t_j + \tau/2).$$

Данная двухслойная четырехточечная схема также формально построена как неявная. Однако из (8.36) можно выразить неизвестное значение u_i^{j+1} через остальные, которые предполагаются известными:

$$u_i^{j+1} = \frac{u_{i-1}^j(1+\lambda) + (u_i^j - u_{i-1}^{j+1})(1-\lambda) + 2\tau\bar{f}_i^j}{1+\lambda}, \quad \lambda = \frac{a\tau}{h}. \tag{8.37}$$

Построенная схема имеет второй порядок точности. Она устойчива на достаточно гладких решениях.

Схема (8.37) получена для случая $a > 0$. Аналогичную ей схему при $a < 0$ можно построить, используя то же самое разностное уравнение (8.36), из которого нужно выразить неизвестное u_{i-1}^{j+1}. Граничные условия задаются здесь в виде (8.33).

Все рассмотренные выше разностные схемы решения линейного уравнения переноса называются *схемами бегущего счета*. Они позволяют последовательно находить значения сеточной функции в узлах разностной сетки.

Схемы бегущего счета, построенные для случая одной пространственной переменной x, можно обобщить на многомерный случай. Рассмотрим для определенности смешанную задачу для двумерного линейного уравнения переноса

$$\frac{\partial U}{\partial t} + a_1 \frac{\partial U}{\partial x} + a_2 \frac{\partial U}{\partial y} = F(x, y, t), \tag{8.38}$$

$$0 \leqslant x \leqslant 1, \quad 0 \leqslant y \leqslant 1, \quad 0 \leqslant t \leqslant T,$$

$$U(x, y, 0) = \Phi(x, y), \tag{8.39}$$

$$U(0, y, t) = \Psi_1(y, t), \quad U(x, 0, t) = \Psi_2(x, t). \tag{8.40}$$

Здесь $a_1 > 0$, $a_2 > 0$ — скорости переноса вдоль осей x, y; (8.39) — начальное условие при $t = 0$; (8.40) — граничные условия при $x = 0$, $y = 0$.

В трехмерной области (x, y, t) построим разностную сетку, ячейки которой имеют форму прямоугольного параллелепипеда. Для этого проведем координатные плоскости через точки деления осей x, y, t:

$$x_i = ih_1 \ (i = 0, 1, \ldots, I),$$

$$y_j = jh_2 \ (j = 0, 1, \ldots, J),$$

$$t_k = k\tau \ (k = 0, 1, \ldots, K).$$

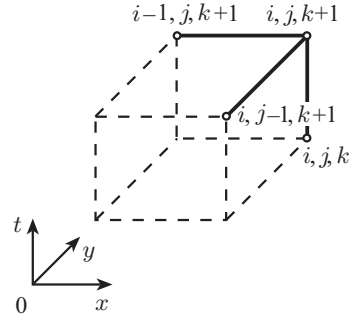


Рис. 8.11. Шаблон для двумерного уравнения

Значение сеточной функции в узле (i, j, k), с помощью которой аппроксимируются значения $U(x_i, y_j, t_k)$, обозначим через u_{ij}^k. Построим безусловно устойчивую разностную схему первого порядка точности, аналогичную схеме (8.35). Шаблон изображен на рис. 8.11, где выделена одна ячейка разностной сетки. Сплошными линиями соединены узлы шаблона. Нижний слой (нижнее основание параллелепипеда) имеет номер k, верхний $k + 1$.

По аналогии с (8.34) запишем разностное уравнение, аппроксимирующее дифференциальное уравнение (8.38):

$$\frac{u_{ij}^{k+1}-u_{ij}^{k}}{\tau}+a_1\frac{u_{ij}^{k+1}-u_{i-1,\,j}^{k+1}}{h_1}+a_2\frac{u_{ij}^{k+1}-u_{i,\,j-1}^{k+1}}{h_2}=f_{ij}^{k}.$$

Разрешим это уравнение относительно значения сеточной функции в узле $(i,j,k+1)$:

$$u_{ij}^{k+1}=\frac{u_{ij}^{k}+\lambda_1 u_{i-1,\,j}^{k+1}+\lambda_2 u_{i,\,j-1}^{k+1}+\tau f_{ij}^{k}}{1+\lambda_1+\lambda_2},$$

$$\lambda_1=a_1\tau/h_1,\quad \lambda_2=a_2\tau/h_2. \tag{8.41}$$

Вычислительный алгоритм этой схемы аналогичен алгоритму одномерной схемы (8.35). Здесь также счет производится по слоям $k=1,2,\ldots,K$. При $k=0$ используется начальное условие (8.39), которое нужно переписать в разностном виде:

$$u_{ij}^{0}=\varphi_{ij}. \tag{8.42}$$

На каждом слое последовательно вычисляются значения сеточной функции в узлах. При этом последовательность перехода от узла к узлу может быть различной: двигаются параллельно либо оси x, либо оси y. Во втором случае последовательность вычисляемых значений следующая: u_{11}^{k+1}, u_{12}^{k+1}, … … , u_{1J}^{k+1}, u_{21}^{k+1}, … , u_{IJ}^{k+1}.

На рис. 8.12 показана нумерация узлов, соответствующая данной последовательности вычислений на каждом временном слое. Точками отмечены расчетные узлы сетки, крестиками — граничные узлы, в которых значения сеточной функции задаются граничными условиями (8.40). Эти условия обходимо записать в сеточном виде:

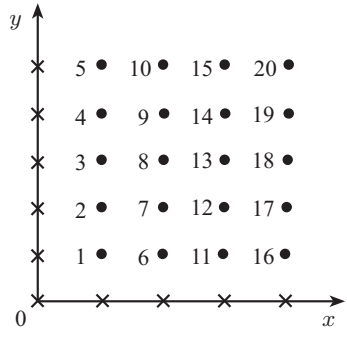


Рис. 8.12. Последовательность вычислений

$$\begin{aligned} u_{0j}^{k+1}&=\psi_1(y_j,t_{k+1}),\\ u_{i0}^{k+1}&=\psi_2(x_i,t_{k+1}) \end{aligned} \tag{8.43}$$

При этом значения u_{00}^{k+1} в угловой точке $(x=0,\,y=0)$ в данной разностной схеме не используются.

Алгоритм решения смешанной задачи (8.38) – (8.40) для двумерного уравнения переноса по схеме (8.41) с учетом сеточных начального и граничных условий (8.42) и (8.43) представлен на рис. 8.13. При этом некоторые блоки (вычисление начальных значений u_{ij}, значений на границе u_{0j}, u_{i0}, пересылка $u_{ij}\to v_{ij}$) даны схематически, хотя каждый из них представляет циклический алгоритм.

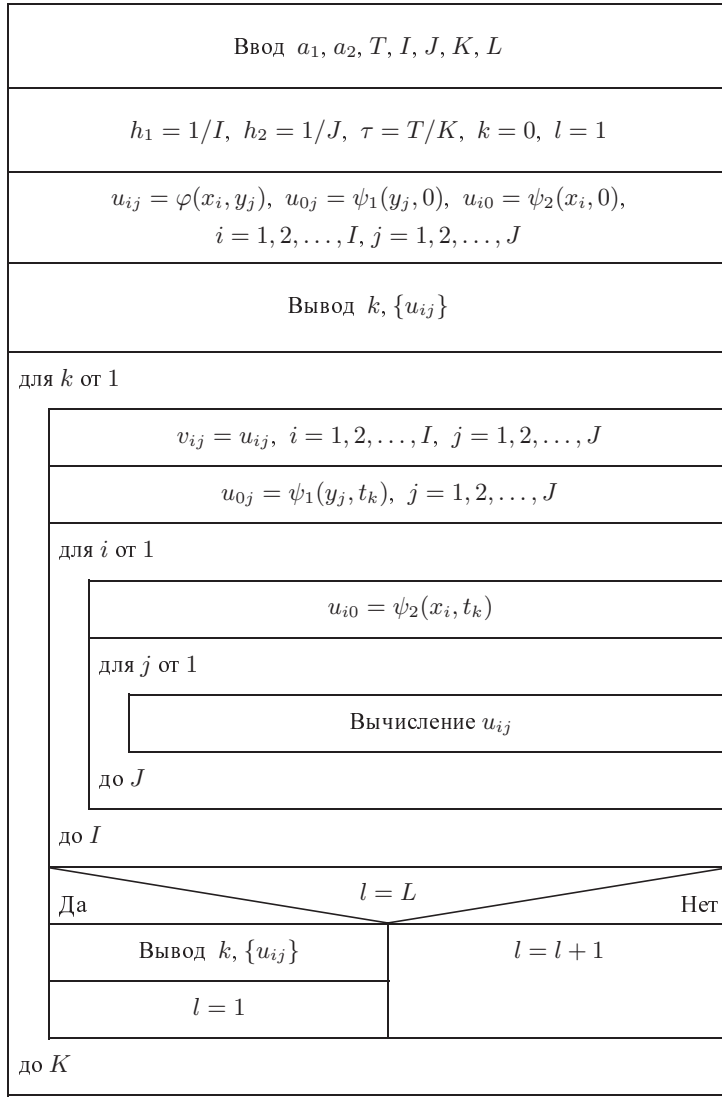


Рис. 8.13. Алгоритм решения двумерного уравнения переноса

В данном алгоритме предусмотрено хранение в памяти машины не полного трехмерного массива искомых значений u_{ij}^k, а лишь значений на двух слоях: v_{ij} — нижний слой, u_{ij} — верхний слой (искомые значения). Введен счетчик выдачи l, решение выдается через каждые L слоев; при $L = 1$ происходит выдача результатов на каждом слое. Блок «Вычисление u_{ij}»

производит вычисление искомого значения по формуле (1), которая в принятых в структурограмме обозначениях имеет вид

$$u_{ij} = \frac{v_{ij} + \lambda_1 u_{i-1,\,j} + \lambda_2 u_{i,\,j-1} + \tau f_{ij}}{1 + \lambda_1 + \lambda_2}\,.$$

2. Квазилинейное уравнение. Разрывные решения. Рассматривая линейное уравнение переноса, мы предполагали, что точное решение задачи является гладкой функцией, причем при построении разностных схем требовалась еще ее дифференцируемость нужное число раз. Сейчас мы будем изучать разрывные решения. Такие решения линейное уравнение переноса может иметь лишь в тех случаях, когда разрывы «заложены» в начальных или граничных условиях.

Рассмотрим теперь *квазилинейные уравнения*, т. е. такие, которые линейны относительно производных искомой функции, однако сама функция может входить в коэффициенты уравнения. Одним из таких уравнений является простейшее квазилинейное уравнение переноса

$$\frac{\partial U}{\partial t} + U\,\frac{\partial U}{\partial x} = 0. \tag{8.44}$$

Это однородное уравнение, т. е. его правая часть равна нулю, что указывает на отсутствие поглощения или источников частиц (энергии). Пусть в начальный момент времени ($t = 0$) решение уравнения (8.44) задано в виде

$$U(x, 0) = U_0(x). \tag{8.45}$$

В уравнении (8.44) роль скорости переноса играет само решение $U(x, t)$. Знак этой функции может быть произвольным, в том числе разным в различных частях расчетной области. Для простоты будем считать, что $U(x, t) > 0$.

Представим уравнение (8.44) в иной форме. Рассмотрим на плоскости (x, t) семейство кривых, определяемых соотношениями

$$x = x(t), \quad dx/dt = U(x, t). \tag{8.46}$$

Вдоль каждой такой кривой функция $U(x, t)$ является сложной функцией одной переменной t: $U = U(x(t), t)$. Полная производная этой функции по t с учетом (8.44), (8.46) есть

$$\frac{dU}{dt} = \frac{\partial U}{\partial t} + \frac{\partial U}{\partial x}\,\frac{dx}{dt} = \frac{\partial U}{\partial t} + U\,\frac{\partial U}{\partial x} = 0.$$

Таким образом, функция U остается постоянной вдоль каждой кривой (8.46). Значение U определяется начальным условием (8.45), взятым в некоторой точке $(x_0, 0)$, через которую проходит кривая:

$$U(x(t), t) = \text{const} = U(x_0, 0) = U_0(x_0). \tag{8.47}$$

Найдем теперь уравнение кривой (8.46), проходящей через точку $(x_0, 0)$. С учетом (8.47) получаем уравнение $dx/dt = U_0(x_0)$, которое легко интегрируется:

$$x = x_0 + U_0(x_0)t. \tag{8.48}$$

Полученное соотношение определяет семейство прямых на плоскости. Функция U не меняется вдоль каждой прямой этого семейства.

Прямые линии (8.48) называются *характеристиками*. Вдоль характеристик уравнения вырождаются в некоторые соотношения между дифференциалами функции, называемые *соотношениями на характеристиках* и имеющими в данном случае вид

$$\frac{dx}{dt} = U, \quad \frac{dU}{dt} = 0.$$

Характеристики (8.48) квазилинейного уравнения (8.44), вообще говоря, не являются параллельными прямыми, как это было в случае линейного уравнения. Если переписать (8.48) в виде $t = (x - x_0)/U_0(x_0)$, то заметим, что тангенс угла наклона характеристик равен $1/U_0(x_0)$. Таким образом, наклон характеристик может меняться в разных точках при $t = 0$. Поэтому, если функция $U_0(x)$ монотонно возрастает, то наклон характеристик слева направо монотонно убывает (веер характеристик). В этом случае решение задачи (8.44), (8.45) однозначно определено, поскольку через каждую точку полуплоскости $t > 0$ проходит одна характеристика, которая переносит в эту точку начальное значение. Такой случай показан на рис. 8.14.

Рассмотрим теперь другой случай. Пусть функция $U_0(x)$ монотонно убывает (или является такой хотя бы на небольшом участке). Тогда

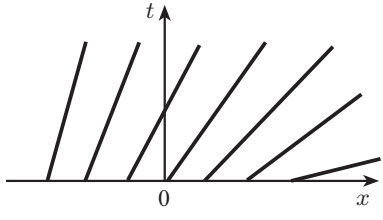


Рис. 8.14

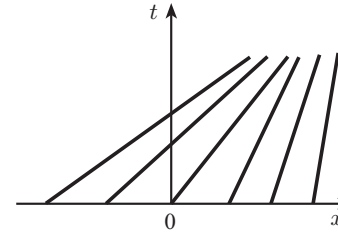


Рис. 8.15

наклон характеристик при движении слева направо увеличивается (рис. 8.15), что приведет к их пересечению. В точке пересечения решение не будет однозначным, поскольку каждая характеристика «принесет» в эту точку свое начальное значение. Поэтому в таких точках решение считается разрывным. Точки разрыва образуют линию разрыва в рассматриваемой области решения.

Различают два вида разрывов: *слабые разрывы*, когда терпят разрыв производные, и *сильные разрывы* — разрывы самого решения. Слабые разрывы в квазилинейном уравнении распространяются по характеристикам, сильные разрывы (в механике сплошных сред это обычно ударные волны) распространяются не по характеристикам. В точках разрыва производные не определены, поэтому уравнение теряет смысл. Следовательно, задачу нужно как-то доопределить, заменив в точках разрыва дифференциальные уравнения некоторыми конечными соотношениями.

Пусть $x = \varphi(t)$ — уравнение линии разрыва, U^- и U^+ — значения решения соответственно слева и справа от точки разрыва, причем $U^- > U^+$ (только в этом случае происходит пересечение характеристик). Тогда значения производной $dx/dt = \varphi'(t)$ на линии разрыва определяют по формуле

$$\varphi'(t) = (U^- + U^+)/2, \quad U^- > U^+. \tag{8.49}$$

Это соотношение на линии разрыва заменяет дифференциальное уравнение. Таким образом, решение задачи (8.44), (8.45), (8.49) ищется в классе разрывных функций.

Перейдем к рассмотрению численных методов решения данной задачи. Они подразделяются на две основные группы: методы с выделением разрывов и методы сквозного счета.

Методы с выделением разрывов являются модификациями рассмотренных выше методов. Различие состоит в том, что во всей области решение ищется обычным способом, а в окрестности линий разрыва счет проводится нестандартным образом. При этом обычно требуется найти сначала точки разрыва, которые к тому же не являются расчетными узлами. Такой естественный способ нахождения разрывных решений отпугивает многих пользователей сложностью алгоритма.

В методах сквозного счета разрыв не выделяется, и весь расчет проводится по единой схеме, что весьма выгодно при организации вычислений на компьютере. Разностные схемы, используемые для таких расчетов, называются *однородными*. Однако в этих схемах разрыв перестает быть разрывом в смысле изменения решения в одной точке. Он растягивается на несколько расчетных узлов, «размазывается». Рассмотрим этот вопрос подробнее.

На рис. 8.16 изображено точное решение U в некоторый момент времени (сплошная линия). В точке x_0 имеется разрыв, причем для простоты значения функции слева (U^-) и справа (U^+) приняты постоянными. При использовании некоторого метода сквозного счета получились значения сеточной функции, отмеченные точками. Мы видим, что сеточная функция является монотонной (в данном случае она не возрастает).

Схемы, которые сохраняют монотонность решения разностной задачи, называются *монотонными*. В теории разностных схем доказывается следующий необходимый и достаточный признак монотонности линейной схемы.

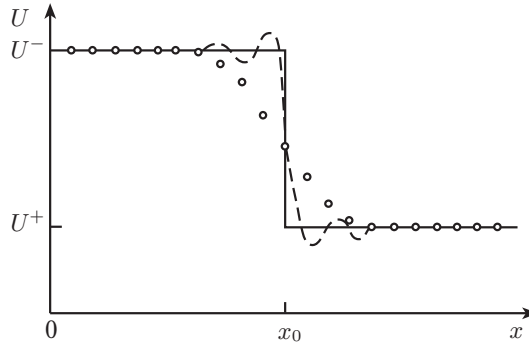


Рис. 8.16. Разрывное решение

Т е о р е м а. *Явная двухслойная разностная схема вида*

$$u_i^{j+1} = \alpha_0 u_{i-k}^j + \alpha_1 u_{i-k+1}^j + \ldots + \alpha_n u_{i-k+n}^j \tag{8.50}$$

монотонна тогда и только тогда, когда $\alpha_0, \alpha_1, \ldots, \alpha_n$ *— неотрицательные числа.*

Можно также показать, что для линейного уравнения переноса такие схемы могут иметь только первый порядок точности. Схемы высших порядков точности не являются монотонными. На рис. 8.16 штриховой линией отмечено решение, которое может быть получено сквозным счетом с использованием схемы второго порядка. Здесь наблюдается нарушение монотонности сеточной функции.

«Размазывание» разрывов решения при переходе от дифференциальной задачи к аппроксимирующей ее разностной схеме объясняется наличием в схеме так называемой *аппроксимационной вязкости*. В частности, схемы (8.28), (8.35) первого порядка точности обладают аппроксимационной вязкостью, а схема второго порядка (8.37) ею не обладает. Понятие аппроксимационной вязкости применимо только для линейных разностных схем вида (8.50).

Одним из приемов, используемых для расчета разрывных решений в рамках нелинейных уравнений (и, в частности, квазилинейных), является введение понятия *искусственной вязкости* (или *псевдовязкости*). Этот прием позволяет превратить разрывные решения в непрерывные и при этом достаточно гладкие. С этой целью в исходное уравнение вводится малая добавка (*возмущение*), и разрывное решение может быть получено как предел введенного гладкого решения при стремлении к нулю параметра возмущения.

Итак, вместо исходного квазилинейного уравнения (8.44) рассмотрим уравнение

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + \frac{\varepsilon^2}{2}\frac{\partial}{\partial x}\left(\frac{\partial U}{\partial x}\right)^2 = 0. \tag{8.51}$$

Здесь последний член в левой части описывает искусственную вязкость, при этом параметр ε мал. Ясно, что при малом значении ε решения уравнений (8.44) и (8.51) при одинаковых начальных условиях будут близкими, если эти решения достаточно гладкие (вторая производная ограничена).

Рассмотрим теперь разрывное решение исходной задачи (8.44), (8.45). Пусть это решение представляет собой ступенчатую функцию (см. рис. 8.16)

$$U = \begin{cases} U^-, & x < at, \\ U^+, & x > at; \end{cases} \tag{8.52}$$

$$a = (U^- + U^+)/2, \quad U^- > U^+. \tag{8.53}$$

Это решение можно трактовать как ударную волну, движущуюся со скоростью a. При этом U^-, U^+ — некоторые постоянные. Легко убедиться, в том, что функция (8.52) удовлетворяет как квазилинейному уравнению (8.44), так и соотношению (8.49), заменяющему на линии разрыва $x = at$ дифференциальное уравнение.

Построим решение уравнения (8.51). Будем искать его в виде

$$U_\varepsilon(x,t) = f(x - at). \tag{8.54}$$

На это решение можно наложить асимптотическое условие, которое состоит в том, что вдали от разрыва решение $U_\varepsilon(x,t)$ уравнения (8.51) и решение $U(x,t)$ уравнения (8.44), являющиеся гладкими функциями, близки, т. е.

$$f(x - at) \to U^\pm, \quad x \to \pm\infty.$$

Подставим решение (8.54) в уравнение (8.51). При этом учтем, что функция $f(x - at)$ является сложной функцией одного аргумента $z = x - at$. Ее производные равны

$$\frac{\partial U}{\partial t} = \frac{df}{dz}\frac{\partial z}{\partial t} = -af', \quad \frac{\partial U}{\partial x} = \frac{df}{dz}\frac{\partial z}{\partial x} = f',$$

$$\frac{\partial}{\partial x}\left(\frac{\partial U}{\partial x}\right)^2 = \frac{\partial}{\partial x}\left(f'^2\right) = 2f'\frac{df'}{dz}\frac{\partial z}{\partial x} = 2f'f''.$$

Подставляя эти выражения в уравнение (8.51), получаем следующее обыкновенное дифференциальное уравнение относительно искомой функции $f(x - at)$:

$$-af' + ff' + \varepsilon^2 f'f'' = 0,$$

или

$$f'(\varepsilon^2 f'' + f - a) = 0.$$

Приравнивая нулю каждый из сомножителей, получаем два значения функции f:

$$f_1 = C_1, \quad f_2 = a + C_2 \sin\frac{x - at}{\varepsilon}, \tag{8.55}$$

где C_1, C_2 — постоянные.

Из значений (8.55) с учетом (8.52), (8.53) можно построить решение, напоминающее «размазанную» ударную волну (рис. 8.17), которое имеет вид

$$U_\varepsilon = \begin{cases} U^-, & \dfrac{x-at}{\varepsilon} \leqslant -\dfrac{\pi}{2}, \\ \dfrac{U^-+U^+}{2} - \dfrac{U^--U^+}{2}\sin\dfrac{x-at}{\varepsilon}, & -\dfrac{\pi}{2} < \dfrac{x-at}{\varepsilon} < \dfrac{\pi}{2}, \\ U^+, & \dfrac{x-at}{\varepsilon} \geqslant \dfrac{\pi}{2}. \end{cases}$$

Это — гладкое решение, оно имеет даже кусочно непрерывную вторую производную. При малом ε зона перехода от U^- к U^+ мала и решение близко к разрывному.

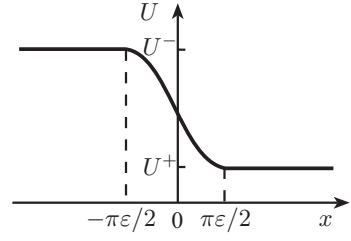


Рис. 8.17. Решение с искусственной вязкостью ($t = 0$)

Таким образом, вместо нахождения разрывного решения задачи (8.44), (8.45), (8.49) можно искать непрерывное, решение уравнения (8.51) при малых значениях ε. А это уравнение решается с помощью однородных разностных схем. В процессе решения следует обратить внимание на выбор шага h (а для неявных схем также τ), с тем, чтобы в области разрыва располагалось хотя бы несколько узлов.

Примером разностной схемы для уравнения (8.51) с искусственной вязкостью может быть следующая схема:

$$\frac{u_i^{j+1}-u_i^j}{\tau} + u_i^j\,\frac{u_i^j-u_{i-1}^j}{h} + \frac{\varepsilon^2}{2}\frac{1}{h}\left[\left(\frac{u_{i+1}^j-u_i^j}{h}\right)^2 - \left(\frac{u_i^j-u_{i-1}^j}{h}\right)^2\right] = 0.$$

Упрощая это выражение и разрешая его относительно неизвестного значения сеточной функции на $(j+1)$-м слое, получаем

$$u_i^{j+1} = u_i^j - \frac{\tau}{h}\,u_i^j(u_i^j-u_{i-1}^j) - \frac{\varepsilon^2\tau}{2h^3}\,(u_{i+1}^j-u_{i-1}^j)(u_{i+1}^j-u_i^j+u_{i-1}^j). \tag{8.56}$$

Эта явная схема условно устойчива при выполнении неравенства

$$\tau \leqslant h/U(x,t),$$

в котором роль скорости распространения возмущения a (для линейного уравнения) играет сама функция U. Разностная схема (8.56) пригодна для решения задач при наличии движущихся разрывов.

3. Консервативные схемы. Для нелинейных уравнений и соответствующих им разностных схем трудно доказывать сходимость. Поэтому пользуются часто так называемым понятием *практической сходимости*. Она состоит в том, что расчеты по данной схеме проводят многократно на сгущающейся сетке. Сходимость к некоторому решению является подтверждением достоверности результатов. Однако такой способ годится только для гладких решений. При решении задач с разрывами сходимость решения к некоторому пределу при $h \to 0$ может оказаться ложной, а получаемое при этом решение — неверным.

Подобных ситуаций можно избежать путем использования *консервативных разностных схем*. Они основаны на дивергентной форме записи исходных уравнений. Поясним суть этой формы. При описании физических процессов исходные уравнения могут записываться в дифференциальной форме относительно искомых функций (например, плотности, давления, скорости и др.). Существует и другая форма записи уравнений, при которой в качестве искомых параметров принимаются масса, энергия, количество движения и т. п., а сами уравнения выражают законы сохранения этих параметров. Такая форма записи уравнений называется *дивергентной*.

Формально квазилинейное уравнение переноса (8.44) можно также записать в дивергентной форме:

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x}\left(\frac{U^2}{2}\right) = 0. \tag{8.57}$$

Проинтегрируем это уравнение по ячейке $x_{i-1} \leqslant x \leqslant x_i,\ t_j \leqslant t \leqslant t_{j+1}$:

$$\int\limits_{x_{i-1}}^{x_i} dx \int\limits_{t_j}^{t_{j+1}} \frac{\partial U}{\partial t}\, dt + \int\limits_{x_{i-1}}^{x_i} dx \int\limits_{t_j}^{t_{j+1}} \frac{\partial}{\partial x}\left(\frac{U^2}{2}\right) dt = 0,$$

или

$$\int\limits_{x_{i-1}}^{x_i} \left[U^{j+1}(x) - U^j(x)\right] dx + \frac{1}{2} \int\limits_{t_j}^{t_{j+1}} \left[U_i^2(t) - U_{i-1}^2(t)\right] dt = 0. \tag{8.58}$$

Здесь $U^j(x) = U(x, t_j)$, $U_i(t) = U(x_i, t)$. Уравнение (8.58) представляет собой точное интегральное уравнение для данной ячейки. Обычно при исследовании физических процессов оно выражает некоторый закон сохранения.

Аналогичное интегральное уравнение можно получить для всей расчетной области $x_0 \leqslant x \leqslant x_I,\ t_0 \leqslant t \leqslant t_J$, если проинтегрировать уравнение (8.57) по этой области:

$$\int\limits_{x_0}^{x_I} \left(U^J - U^0\right) dx + \frac{1}{2} \int\limits_{t_0}^{t_J} \left(U_I^2 - U_0^2\right) dt = 0. \tag{8.59}$$

Уравнения (8.58) и (8.59) содержат лишь функции, относящиеся к границе соответственно ячейки и всей расчетной области. Эти уравнения можно трактовать как физические законы сохранения: сколько массы, энергии и т. д. в область через границу втекло, столько и вытекло. При этом, если просуммировать уравнение (8.58) по всем ячейкам, получается уравнение (8.59) для всей области. Таким образом, из законов сохранения по каждой ячейке следует закон сохранения для всей области. Схемы, не обладающие этим свойством, называются *неконсервативными*. При их суммировании по всем ячейкам появляется некоторая величина, называемая *дисбалансом*, которая приводит к нарушению закона сохранения для всей области.

В консервативных схемах дисбаланс равен нулю. Приведем пример построения такой схемы. Для этого нужно использовать некоторый численный метод вычисления интегралов, входящих в уравнение (8.58). Воспользуемся для простоты формулой прямоугольников, причем узлы интегрирования предполагаем совпадающими с узлами рассматриваемой разностной сетки. Окончательно получим разностную схему вида

$$\frac{u_i^{j+1} - u_i^j}{\tau} + \frac{(u_i^j)^2 - (u_{i-1}^j)^2}{2h} = 0.$$

Отсюда можно найти значение искомой функции на верхнем слое с помощью решения на нижнем слое. Следовательно, это явная схема. Она обладает свойством консервативности. Аналогичным образом, выбирая различные шаблоны, можно построить другие консервативные разностные схемы.

Консервативные схемы дают результаты с хорошей точностью как для разрывных, так и непрерывных решений. Они оказались полезными при исследовании различных физических явлений. Конкретную схему для решения данной задачи выбирают с учетом требований этой задачи, предъявляемых к схеме (монотонность схемы, однородность, порядок аппроксимации и др.), которые часто бывают противоречивы. Выбранная схема должна быть испытана на решении тестовых задач.

4. Системы уравнений. Характеристики. Для решения систем уравнений с частными производными первого порядка могут быть использованы различные разностные схемы метода сеток, разработанные для одного уравнения. С этой целью формально систему уравнений можно записать в векторной форме с помощью одного уравнения, и тогда вид разностных формул сохраняется таким же, как и для скалярного уравнения. Разница состоит в том, что вместо скалярной сеточной функции вводится векторная.

Рассмотрим систему двух квазилинейных уравнений относительно искомых функций $U(x,t)$, $V(x,t)$:

$$\begin{aligned} a_{11}\frac{\partial U}{\partial t} + a_{12}\frac{\partial V}{\partial t} + b_{11}\frac{\partial U}{\partial x} + b_{12}\frac{\partial V}{\partial x} &= F_1(x,t,U,V), \\ a_{21}\frac{\partial U}{\partial t} + a_{22}\frac{\partial V}{\partial t} + b_{21}\frac{\partial U}{\partial x} + b_{22}\frac{\partial V}{\partial x} &= F_2(x,t,U,V). \end{aligned} \tag{8.60}$$

Коэффициенты a_{mn}, b_{mn} $(m, n = 1, 2)$ этой системы переменные и зависят от x, t, U, V. Введем следующие обозначения: $\mathbf{U}$ — искомый вектор; $\mathbf{F}$ — вектор правой части; A, B — матрицы коэффициентов:

$$\mathbf{U} = \begin{pmatrix} U \\ V \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

Запишем систему уравнений (8.60) в векторном виде:

$$A\frac{\partial \mathbf{U}}{\partial t} + B\frac{\partial \mathbf{U}}{\partial x} = \mathbf{F}.$$

Для решения этого квазилинейного векторного уравнения могут быть использованы различные разностные схемы, которые применяются для решения одного уравнения.

Мы не будем повторять сказанное ранее для одного уравнения, а остановимся на одном частном случае системы (8.60), важном для приложений. Речь идет о системах гиперболического типа. Введем матрицу $C = A\alpha - B\beta$ где α, β — некоторые числа. Тогда определитель этой матрицы

$$\det C = \begin{vmatrix} a_{11}\alpha - b_{11}\beta & a_{12}\alpha - b_{12}\beta \\ a_{21}\alpha - b_{21}\beta & a_{22}\alpha - b_{22}\beta \end{vmatrix} \tag{8.61}$$

является *квадратичной формой* относительно α, β, т. е.

$$\det C = Q(\alpha, \beta) = q_1\alpha^2 + q_2\alpha\beta + q_3\beta^2, \tag{8.62}$$

где коэффициенты q_1, q_2, q_3, легко выразить через элементы матриц A, B, раскрывая определитель (8.61).

Система уравнений с частными производными первого порядка (8.60) называется *гиперболической*, если квадратичная форма (8.62) разлагается на вещественные линейные множители:

$$Q(\alpha, \beta) = (\nu_1\alpha - \mu_1\beta)(\nu_2\alpha - \mu_2\beta),$$

причем векторы $\{\mu_1, \nu_1\}$, $\{\mu_2, \nu_2\}$ неколлинеарны. Эти векторы в каждой точке плоскости (x, t) образуют два направления, которые называются *характеристическими*. Линия, касательная к которой в каждой точке имеет характеристическое направление, называется *характеристикой*. Через каждую точку проходят две характеристики, соответствующие двум характеристическим направлениям. Таким образом, всю плоскость (x, t) можно покрыть двумя семействами характеристик (рис. 8.18).

Заметим, что в случае системы уравнений (8.60) с постоянными коэффициентами характеристические направления, если они существуют, постоянны для всех точек плоскости. Им соответствуют два семейства прямолинейных характеристик. В самом общем случае, когда коэффициенты системы (8.60) зависят от x, t, U, V, характеристики могут существовать в одной части плоскости (x, t) и отсутствовать в другой. Следовательно,

гиперболичность системы (8.60) может быть не на всей плоскости, а лишь в некоторой области.

Наряду с гиперболическими системами существуют также *параболические* (с одним семейством характеристик) и *эллиптические* (действительных характеристик нет) системы.

Характеристики можно использовать для построения алгоритма численного решения системы уравнений с частными производными в области ее гиперболичности. Такой способ решения называется *методом характеристик*.

Не приводя подробных выкладок и опуская сами формулы, изложим идею метода характеристик. Рассмотрим задачу Коши. Пусть при $t = 0$ заданы начальные значения функций $U(x)$, $V(x)$. Выбираем любой отрезок $[a, b]$ на оси x и разбиваем его на части точками A_0, A_1, ..., A_n (рис. 8.19). В данном случае принято $n = 4$.

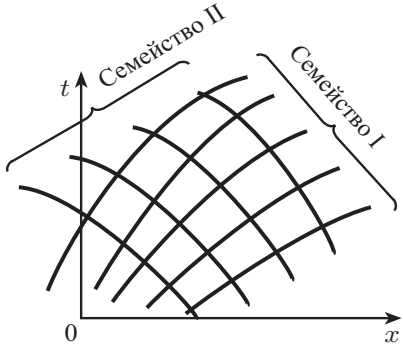


Рис. 8.18. Характеристики

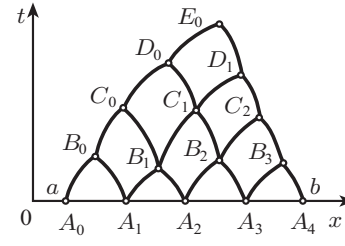


Рис. 8.19

Из точки A_0 проводим характеристику первого семейства, из A_1 — второго. Находим точку пересечения B_0. Используя некоторые соотношения (характеристические) вдоль отрезков характеристик A_0B_0 и A_1B_0, заменяющие исходные уравнения, вычисляем искомые функции в точке B_0. Аналогично находим решение в других точках слоя B. При этом в отличие от метода сеток этот слой не является прямолинейным отрезком $t = \text{const}$, а определяется точками пересечения характеристик.

Далее вычисляем искомые значения в точках слоев C, D и т. д. При этом каждый раз (при решении задачи Коши) при переходе от слоя к слою число узлов уменьшается на единицу, так что на последнем слое получается лишь один узел. Область решения задачи Коши представляет собой криволинейный треугольник с кусочно гладкими сторонами.

При решении краевой задачи используются значения искомых функций на границах. В этом случае расчетная область изменяется: она прилегает к границе $x = \text{const}$, на которой заданы значения функций $U(x)$, $V(x)$. При этом вблизи границы используются характеристики одного семейства,

выходящие из границы и попадающие в расчетную область. Если граничные условия задаются при двух значениях x, то алгоритм метода характеристик значительно усложняется.

Достоинством метода характеристик является то, что он основан на физической сущности задачи, поскольку возмущения распространяются по характеристикам. Метод позволяет выявить разрывы в решении. Недостатком метода является нерегулярность получаемой сетки, поскольку узлы располагаются неравномерно (в точках пересечения характеристик).

Для устранения этого недостатка разработаны так называемые *сеточно-характеристические методы*. Их идея состоит в том, что сетка фиксируется заранее, а характеристики проводятся «назад» из узлов $(j+1)$-го слоя до пересечения с j-м слоем. Значения U, V в точках пересечения вычисляются путем интерполяции по ранее найденному решению в узлах j-го слоя.

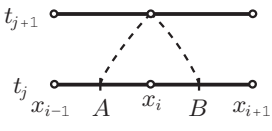


Рис. 8.20.

Геометрическая интерпретация сеточно-характеристического метода показана на рис. 8.20. Здесь точками отмечены заранее выбранные узлы; штриховые линии — отрезки характеристик. Значения функций в точках пересечения A и B находятся интерполированием решения в узлах $(i-1,j)$, (i,j) и $(i+1,j)$. Эти значения используются для определения решения в расчетном узле $(i,j+1)$.

§ 3. Уравнения второго порядка

1. Волновое уравнение. Одним из наиболее распространенных в инженерной практике уравнений с частными производными второго порядка является волновое уравнение, описывающее различные виды колебаний. Поскольку колебания — процесс нестационарный, то одной из независимых переменных является время t. Кроме того, независимыми переменными в уравнении являются также пространственные координаты x, y, z. В зависимости от их количества различают одномерное, двумерное и трехмерное волновые уравнения.

Одномерное волновое уравнение описывает продольные колебания стержня, сечения которого совершают плоскопараллельные колебательные движения, а также поперечные колебания тонкого стержня (струны) и другие задачи. *Двумерное волновое уравнение* используется для исследования колебаний тонкой пластины (мембраны). *Трехмерное волновое уравнение* описывает распространение волн в пространстве (например, звуковых волн в жидкости, упругих волн в сплошной среде и т. п.).

Рассмотрим одномерное волновое уравнение, которое можно записать в виде

$$\frac{\partial^2 U}{\partial t^2} = a^2 \frac{\partial^2 U}{\partial x^2}. \tag{8.63}$$

Для поперечных колебаний струны искомая функция $U(x,t)$ описывает положение струны в момент t. В этом случае $a^2 = T/\rho$, где T — натяжение струны, ρ — ее линейная (погонная) плотность. Колебания предполагаются малыми, т. е. амплитуда мала по сравнению с длиной струны. Кроме того, уравнение (8.63) записано для случая свободных колебаний. В случае вынужденных колебаний в правой части уравнения добавляется некоторая функция $f(x,t)$, характеризующая внешние воздействия. Сопротивление среды колебательному процессу не учитывается.

Простейшей задачей для уравнения (8.63) является задача Коши: в начальный момент времени задаются два условия (количество условий равно порядку входящей в уравнение производной по t):

$$U\big|_{t=0} = U(x,0) = \varphi(x), \quad \partial U/\partial t\big|_{t=0} = \psi(x). \tag{8.64}$$

Эти условия описывают начальную форму струны $U = \varphi(x)$ и скорость ее точек $\psi(x)$.

На практике чаще приходится решать не задачу Коши для бесконечной струны, а смешанную задачу для ограниченной струны некоторой длины l. В этом случае задают граничные условия на ее концах. В частности, при закрепленных концах их смещения равны нулю, и граничные условия имеют вид

$$U\big|_{x=0} = 0, \quad U\big|_{x=l} = 0. \tag{8.65}$$

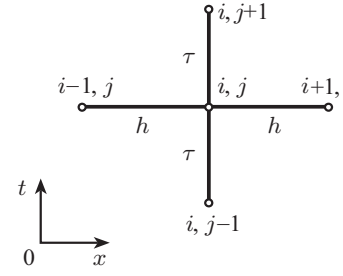


Рис. 8.21. Шаблон явной схемы

Рассмотрим некоторые разностные схемы для решения задачи (8.63) – (8.65). Простейшей является явная трехслойная схема типа крест (шаблон показан на рис. 8.21). Заменим в уравнении (8.63) вторые производные искомой функции U по t и x их конечно-разностными соотношениями с помощью значений сеточной функции u_i^j в узлах сетки (x_i, t_j):

$$\frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{\tau^2} = a^2 \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2},$$
$$i = 1, 2, \ldots, I-1, \quad j = 1, 2, \ldots, J-1.$$

Отсюда можно найти явное выражение для значения сеточной функции на $(j+1)$-м слое:

$$u_i^{j+1} = 2(1-\lambda)u_i^j + \lambda(u_{i+1}^j + u_{i-1}^j) - u_i^{j-1}, \quad \lambda = a^2\tau^2/h^2. \tag{8.66}$$

Здесь, как обычно в трехслойных схемах, для определения неизвестных значений на $(j+1)$-м слое нужно знать решения на j-м и $(j-1)$-м

слоях. Поэтому начать счет по формулам (8.66) можно лишь для второго слоя, а решения на нулевом и первом слоях должны быть известны. Они находятся с помощью начальных условий (8.64).

На нулевом слое имеем

$$u_i^0 = \varphi(x_i), \quad i = 0, 1, \ldots, I. \tag{8.67}$$

Для получения решения на первом слое воспользуемся вторым начальным условием (8.64). Производную $\partial U/\partial t$ заменим конечно-разностной аппроксимацией. В простейшем случае полагают

$$\left.\frac{\partial U}{\partial t}\right|_{\substack{t=0\\x=x_i}} = \psi(x_i) \approx \frac{u_i^1 - u_i^0}{\tau}. \tag{8.68}$$

Из этого соотношения можно найти значения сеточной функции на первом временном слое:

$$u_i^1 = u_i^0 + \tau\psi(x_i), \quad i = 0, 1, \ldots, I. \tag{8.69}$$

Отметим, что аппроксимация начального условия в виде (8.68) ухудшает аппроксимацию исходной дифференциальной задачи: погрешность аппроксимации становится порядка $O(h^2 + \tau)$, т. е. первого порядка по τ, хотя сама схема (8.66) имеет второй порядок аппроксимации по h и τ. Положение можно исправить, если вместо (8.69) взять более точное представление

$$u_i^1 = u_i^0 + \tau \left.\frac{\partial U}{\partial t}\right|_{\substack{t=0\\x=x_i}} + \frac{\tau^2}{2} \left.\frac{\partial^2 U}{\partial t^2}\right|_{\substack{t=0\\x=x_i}}. \tag{8.70}$$

Вместо $\partial U/\partial t$ нужно взять $\psi(x)$. А выражение для второй производной можно найти с использованием исходного уравнения (8.63) и первого начального условия (8.64). Получим

$$\left.\frac{\partial^2 U}{\partial t^2}\right|_{t=0} = a^2 \left.\frac{\partial^2 U}{\partial x^2}\right|_{t=0} = a^2 \frac{\partial^2 \varphi}{\partial x^2}.$$

Тогда (8.70) принимает вид

$$u_i^1 = u_i^0 + \tau\psi(x_i) + \frac{a^2\tau^2}{2}\varphi''(x_i), \quad i = 0, 1, \ldots, I. \tag{8.71}$$

Разностная схема (8.66) с учетом (8.71) обладает погрешностью аппроксимации порядка $O(h^2 + \tau^2)$.

При решении смешанной задачи с граничными условиями вида (8.65), т. е. когда на концах рассматриваемого отрезка заданы значения самой функции, второй порядок аппроксимации сохраняется. В этом случае для удобства крайние узлы сетки располагают в граничных точках ($x_0 = 0$, $x_I = l$). Однако граничные условия могут задаваться и для производной.

Например, в случае свободных продольных колебаний стержня на его незакрепленном конце задается условие

$$\frac{\partial U}{\partial t}\Big|_{x=l} = 0. \qquad (8.72)$$

Если это условие записать в разностном виде с первым порядком аппроксимации, то погрешность аппроксимации схемы станет порядка $O(h+\tau^2)$. Поэтому для сохранения второго порядка данной схемы по h необходимо граничное условие (8.72) аппроксимировать со вторым порядком.

Рассмотренная разностная схема (8.66) решения задачи (8.63) — (8.65) условно устойчива. Необходимое и достаточное условие устойчивости имеет вид

$$\frac{a\tau}{h} < 1. \qquad (8.73)$$

Следовательно, при выполнении этого условия и с учетом аппроксимации схема (8.66) сходится к исходной задаче со скоростью $O(h^2+\tau^2)$. Данная схема часто используется в практических расчетах. Она обеспечивает приемлемую точность получения решения $U(x,t)$, которое имеет непрерывные производные четвертого порядка.

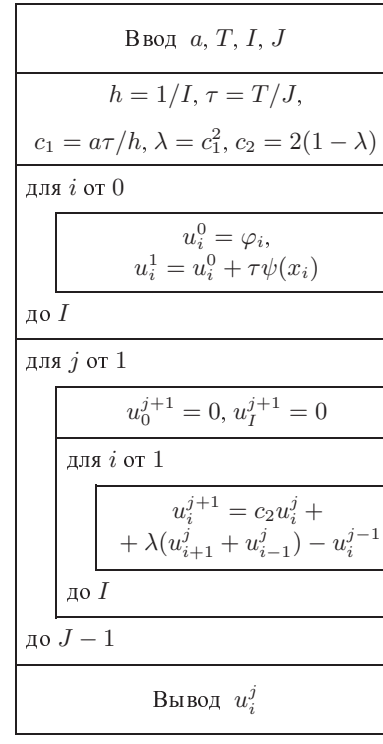


Рис. 8.22. Алгоритм решения волнового уравнения

Алгоритм решения задачи (8.63) – (8.65) с помощью данной явной разностной схемы приведен на рис. 8.22. Здесь представлен простейший вариант, когда все значения сеточной функции, образующие двумерный массив, по мере вычисления хранятся в памяти компьютера, а после решения задачи происходит вывод результатов. Можно было бы предусмотреть хранение решения лишь на трех слоях, что сэкономило бы память. Вывод результатов в таком случае можно производить в процессе счета (см. рис. 8.13).

Существуют и другие разностные схемы решения волнового уравнения. В частности, иногда удобнее использовать неявные схемы, чтобы избавиться от ограничений на величину шага, налагаемых условием (8.73). Эти схемы обычно абсолютно устойчивы, однако алгоритм решения задачи и программа для компьютера усложняются.

Построим простейшую неявную схему. Вторую производную по t в уравнении (8.63) аппроксимируем, как и ранее, по трехточечному шаблону с помощью значений сеточной функции на слоях $j-1$, j, $j+1$. Производную до x заменяем полусуммой ее аппроксимации на $(j+1)$-м и $(j-1)$-м слоях (рис. 8.23):

$$\frac{u_i^{j+1}-2u_i^j+u_i^{j-1}}{\tau^2}=\frac{a^2}{2}\left(\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{h^2}+\frac{u_{i+1}^{j-1}-2u_i^{j-1}+u_{i-1}^{j-1}}{h^2}\right).$$

Из этого соотношения можно получить систему уравнений относительно неизвестных значений сеточной функции на $(j+1)$-м слое:

$$\lambda u_{i-1}^{j+1}-(1+2\lambda)u_i^{j+1}+\lambda u_{i+1}^{j+1}=(1+2\lambda)u_i^{j-1}-\lambda(u_{i+1}^{j-1}+u_{i-1}^{j-1})-2u_i^j, \tag{8.74}$$

$$\lambda=a^2\tau^2/h^2,\quad i=1,2,\ldots,I-1,\quad j=1,2,\ldots,J-1.$$

Полученная неявная схема устойчива и сходится со скоростью $O(h^2+\tau^2)$. Систему линейных алгебраических уравнений (8.74) можно, в частности, решать методом прогонки. К этой системе следует добавить разностные начальные и граничные условия. Так выражения (8.67), (8.69) или (8.71) могут быть использованы для вычисления значений сеточной функции на нулевом и первом слоях по времени.

При наличии двух или трех независимых пространственных переменных волновые уравнения принимают вид

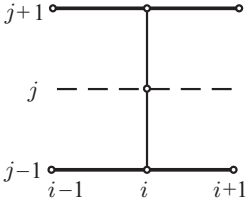


Рис. 8.23. Шаблон неявной схемы

$$\frac{\partial^2U}{\partial t^2}=a^2\left(\frac{\partial^2U}{\partial x^2}+\frac{\partial^2U}{\partial y^2}\right),$$

$$\frac{\partial^2U}{\partial t^2}=a^2\left(\frac{\partial^2U}{\partial x^2}+\frac{\partial^2U}{\partial y^2}+\frac{\partial^2U}{\partial z^2}\right).$$

Для них также могут быть построены разностные схемы по аналогии с одномерным волновым уравнением. Разница состоит в том, что нужно аппроксимировать производные по двум или трем пространственным переменным, что, естественно, усложняет алгоритм и требует значительно больших объемов памяти и времени счета. Подробнее двумерные задачи будут рассмотрены ниже для уравнения теплопроводности.

2. Уравнение теплопроводности. Ранее (см. § 1, пп. 2, 3) уже были построены и исследованы разностные схемы решения смешанной задачи для одномерного уравнения теплопроводности:

$$\frac{\partial U}{\partial t}=a\,\frac{\partial^2U}{\partial x^2},\quad 0\leqslant x\leqslant 1,\quad t>0,\quad a>0,$$
$$U(x,0)=\varphi(x),\quad U(0,t)=\psi_1(t),\quad U(1,t)=\psi_2(t), \tag{8.75}$$

Были получены две двухслойные схемы — явная (8.3) и неявная (8.4). В явной схеме значения сеточной функции u_i^{j+1} на верхнем $(j+1)$-м слое вычислялись через решение на нижнем слое с помощью соотношения (8.13). Для нахождения решения на $(j+1)$-м слое по неявной схеме была получена трехдиагональная система линейных алгебраических уравнений (8.22), которая может быть решена методом прогонки.

Неявная схема безусловно устойчива, явная схема устойчива при выполнении условия

$$a\tau/h^2 \leqslant 1/2.$$

Обе схемы сходятся к решению исходной задачи со скоростью $O(h^2+\tau)$.

Схемы (8.3), (8.4) построены для случая, когда значения искомой функции (температуры) U на границах $x=0$, $x=1$ определяются заданными функциями $\psi_1(t)$, $\psi_2(t)$. Однако граничные условия в смешанной задаче (8.75) могут быть и иными, в них может входить производная искомой функции. Например, если конец стержня $x=0$ теплоизолирован, то условие имеет вид

$$\frac{\partial U}{\partial x}\Big|_{x=0} = 0.$$

В этом случае, как и при решении волнового уравнения, данное условие нужно записывать в схемах (8.3), (8.4) в разностном виде.

Перейдем теперь к построению разностных схем для уравнения теплопроводности с двумя пространственными переменными. Положим для простоты $a=1$. Тогда это уравнение можно записать в виде

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\,. \tag{8.76}$$

Пусть при $t=0$ начальное условие задано в виде

$$U(x,y,0) = \varphi(x,y). \tag{8.77}$$

В отличие от волнового уравнения, требующего два начальных условия, в уравнение теплопроводности входит только первая производная по t, и необходимо задавать одно начальное условие.

Часто задачи теплопроводности или диффузии, описываемые двумерным уравнением (8.76), решаются в ограниченной области. Тогда, кроме начального условия (8.77), нужно формулировать граничные условия. В частности, если расчетная область представляет прямоугольный параллелепипед $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$, $0 \leqslant t \leqslant T$ (рис. 8.24), то нужно задавать граничные условия на его боковых гранях. Начальное условие (8.77) задано на нижнем основании параллелепипеда.

Введем простейшую сетку с ячейками в виде прямоугольных параллелепипедов, для чего проведем три семейства плоскостей: $x_i = ih_1$ $(i = 0,1,\ldots,I)$, $y_j = jh_2$ $(j=0,1,\ldots,J)$, $t_k = k\tau$ $(k=0,1,\ldots,K)$. Значение сеточной функции в узлах (x_i, y_j, t_k) обозначим символом u_{ij}^k. Используя эти значения, можно построить разностные схемы для уравнения (8.76).

Рассмотренные выше схемы для одномерного уравнения легко обобщаются на двумерный случай.

Построим явную разностную схему, шаблон которой изображен на

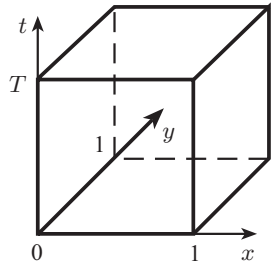


Рис. 8.24. Расчетная область

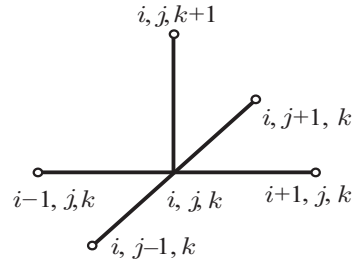


Рис. 8.25. Шаблон двумерной схемы

рис. 8.25. Аппроксимируя производные отношениями конечных разностей, получаем следующее сеточное уравнение:

$$\frac{u_{ij}^{k+1}-u_{ij}^{k}}{\tau}=\frac{u_{i+1,\,j}^{k}-2u_{ij}^{k}+u_{i-1,\,j}^{k}}{h_1^2}+\frac{u_{i,\,j+1}^{k}-2u_{ij}^{k}+u_{i,\,j-1}^{k}}{h_2^2}\,.$$

Отсюда можно найти явное выражение для значения сеточной функции на $(k+1)$-м слое:

$$u_{ij}^{k+1}=(1-2\lambda_1-2\lambda_2)u_{ij}^{k}+\lambda_1(u_{i+1,\,j}^{k}+u_{i-1,\,j}^{k})+\lambda_2(u_{i,\,j+1}^{k}+u_{i,\,j-1}^{k}),$$
$$\lambda_1=\tau/h_1^2,\quad \lambda_2=\tau/h_2^2. \tag{8.78}$$

Условие устойчивости имеет вид

$$\lambda_1+\lambda_2=\tau/h_1^2+\tau/h_2^2\leqslant 1/2. \tag{8.79}$$

При $\lambda_1+\lambda_2=1/2$ получается особенно простой вид схемы (8.78):

$$u_{ij}^{k+1}=\lambda_1(u_{i+1,\,j}^{k}+u_{i-1,\,j}^{k})+\lambda_2(u_{i,\,j+1}^{k}+u_{i,\,j-1}^{k}). \tag{8.80}$$

Полученная схема сходится со скоростью $O(h_1^2+h_2^2+\tau)$.

Формулы (8.78) или (8.80) представляют собой рекуррентные соотношения для последовательного вычисления сеточной функции во внутренних узлах слоев $k=1,2,\ldots,K$. На нулевом слое используется начальное условие (8.77), которое записывается в виде

$$u_{ij}^{0}=\varphi(x_i,y_j). \tag{8.81}$$

Значения u_{0j}^{k}, u_{Ij}^{k}, u_{i0}^{k}, u_{iJ}^{k} в граничных узлах вычисляются с помощью граничных условий.

Алгоритм решения смешанной задачи для двумерного уравнения теплопроводности изображен на рис. 8.26. Здесь решение хранится на двух слоях: нижнем (массив v_{ij}) и верхнем (массив u_{ij}). Блоки граничных условий необходимо сформировать в зависимости от конкретного вида этих условий. Вывод результатов производится на каждом слое, хотя можно ввести шаг выдачи (см. рис. 8.13).

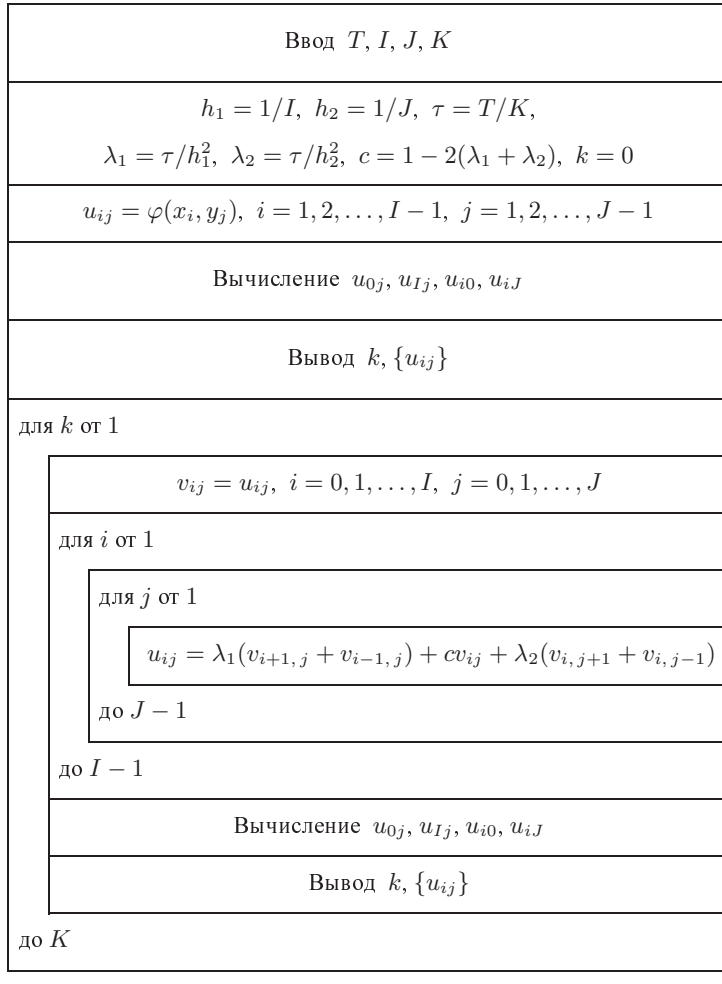


Рис. 8.26. Алгоритм решения двумерного уравнения теплопроводности

Построим теперь абсолютно устойчивую неявную схему для решения уравнения (8.76), аналогичную схеме (8.4) для одномерного уравнения теплопроводности. Аппроксимируя в (8.76) вторые производные по

пространственным переменным на $(k+1)$-м слое, получаем следующее разностное уравнение:

$$\frac{u_{ij}^{k+1}-u_{ij}^{k}}{\tau}=\frac{u_{i+1,\,j}^{k+1}-2u_{ij}^{k+1}+u_{i-1,\,j}^{k+1}}{h_1^2}+\frac{u_{i,\,j+1}^{k+1}-2u_{ij}^{k+1}+u_{i,\,j-1}^{k+1}}{h_2^2}. \tag{8.82}$$

Это уравнение можно записать в виде системы линейных алгебраических уравнений относительно значений сеточной функции на каждом слое:

$$\lambda_1(u_{i-1,\,j}^{k+1}+u_{i+1,\,j}^{k+1})-(1+2\lambda_1+2\lambda_2)u_{ij}^{k+1}+\lambda_2(u_{i,\,j-1}^{k+1}+u_{i,\,j+1}^{k+1})=-u_{ij}^{k}, \tag{8.83}$$

$$\lambda_1=\tau/h_1^2,\quad \lambda_2=\tau/h_2^2,$$
$$i=1,2,\ldots,I-1,\quad j=1,2,\ldots,J-1.$$

К этой системе уравнений нужно добавить граничные условия для определения значений сеточной функции в граничных узлах (т. е. при $i=0,I$; $j=0,J$). На нулевом слое решение находится из начального условия (8.77), представленного в виде (8.81).

Система (8.83), полученная для двумерного уравнения теплопроводности, имеет более сложный вид, чем аналогичная система (8.22) для одномерного случая, которую можно решить методом прогонки. Таким образом, распространение неявной схемы на многомерный случай приводит к значительному усложнению вычислительного алгоритма и увеличению объема вычислений.

Недостатком явной схемы (8.78) является жесткое ограничение на шаг по времени τ, вытекающее из условия (8.79). Существуют абсолютно устойчивые экономичные разностные схемы, позволяющие вести расчет со сравнительно большим значением шага по времени $(\tau\sim h)$ и требующие меньшего объема вычислений. Две из них будут рассмотрены в п. 3

3. Понятие о схемах расщепления. Основой построения рассматриваемых схем является разбиение расчета на одном шаге по времени, т. е. перехода от k-го к $(k+1)$-му слою на отдельные этапы. Такие схемы называют *схемами расщепления* или *схемами дробных шагов*. Они сохраняют преимущества как явных схем (простой вычислительный алгоритм), так и неявных (возможность счета с большими значениями шага по времени) и лишены присущих этим схемам недостатков.

Одной из таких схем, используемых для решения задач при наличии двух пространственных переменных, является *схема переменных направлений* (в литературе можно встретить также название *продольно-поперечная схема*). Суть этой схемы состоит в том, что шаг по времени τ делится на два полушага. Первый из них проводится со слоя k до промежуточного слоя $t=t_k+\tau/2$, который обозначается полуцелым индексом $k+1/2$. Второй полушаг проводится со слоя $k+1/2$ до слоя $k+1$.

На первом полушаге вторая производная по одной из координат, например $\partial^2 U/\partial x^2$, аппроксимируется на слое $k+1/2$, а вторая производная по другой координате, $\partial^2 U/\partial y^2$, — на слое k:

$$\frac{u_{ij}^{k+1/2}-u_{ij}^{k}}{\tau/2}=\frac{u_{i+1,\,j}^{k+1/2}-2u_{ij}^{k+1/2}+u_{i-1,\,j}^{k+1/2}}{h_1^2}+\frac{u_{i,\,j+1}^{k}-2u_{ij}^{k}+u_{i,\,j-1}^{k}}{h_2^2}. \tag{8.84}$$

Получившееся разностное уравнение приводит к неявной схеме для нахождения значений $u_{ij}^{k+1/2}$. На втором полушаге, наоборот, приводящая к неявной схеме аппроксимация используется только по направлению y, т. е. $\partial^2 U/\partial y^2$ аппроксимируется на слое $k+1$, а $\partial^2 U/\partial x^2$ — по-прежнему на слое $k+1/2$:

$$\frac{u_{ij}^{k+1}-u_{ij}^{k+1/2}}{\tau/2}=\frac{u_{i+1,\,j}^{k+1/2}-2u_{ij}^{k+1/2}+u_{i-1,\,j}^{k+1/2}}{h_1^2}+\frac{u_{i,\,j+1}^{k+1}-2u_{ij}^{k+1}+u_{i,\,j-1}^{k+1}}{h_2^2}. \tag{8.85}$$

Таким образом, вместо разностного уравнения (8.82) в чисто неявной схеме мы получили два уравнения, каждое из которых, по существу, соответствует неявной схеме по одному из координатных направлений.

Уравнения (8.84), (8.85) можно переписать в виде систем линейных алгебраических уравнений относительно значений искомых функций соответственно в узлах $(k+1/2)$-го и $(k+1)$-го слоев:

$$\lambda_1 u_{i-1,\,j}^{k+1/2}-(1+2\lambda_1)u_{ij}^{k+1/2}+\lambda_1 u_{i+1,\,j}^{k+1/2}= \\ =(2\lambda_2-1)u_{ij}^{k}-\lambda_2(u_{i,\,j+1}^{k}+u_{i,\,j-1}^{k}), \tag{8.86}$$

$$\lambda_2 u_{i,\,j-1}^{k+1}-(1+2\lambda_2)u_{ij}^{k+1}+\lambda_2 u_{i,\,j+1}^{k+1}= \\ =(2\lambda_1-1)u_{ij}^{k+1/2}-\lambda_1(u_{i+1,\,j}^{k+1/2}+u_{i-1,\,j}^{k+1/2}), \tag{8.87}$$

$$\lambda_1=\tau/(2h_1^2),\quad \lambda_2=\tau/(2h_2^2),\quad i=1,2,\ldots,I-1,\quad j=1,2,\ldots,J-1.$$

К этим системам уравнений необходимо добавить начальные условия в виде (8.81), а также граничные условия на каждом из этих дробных по времени шагов.

Матрицы систем (8.86) и (8.87) трехдиагональные, и для решения этих систем может быть использован метод прогонки. При этом сначала необходимо решить систему уравнений (8.86), из которой находятся значения сеточной функции $u_{ij}^{k+1/2}$. Эти значения используются затем для вычисления искомых значений u_{ij}^{k+1} из системы (8.87).

Заметим, что диагональные элементы матриц систем (8.86) и (8.87) преобладают, поэтому выполняются условия устойчивости прогонки.

Это также обеспечивает существование и единственность решения данных систем, т. е. разностного решения. Приведенная схема переменных направлений безусловно устойчива, она сходится со скоростью $O(h_1^2 + h_2^2 + \tau^2)$.

Как уже отмечалось, рассмотренная схема весьма эффективна для случая двух пространственных переменных. Однако на случай трех и более переменных она непосредственно не обобщается.

Рассмотрим другой тип схем — *локально-одномерные схемы*. Их построение основано на введении на каждом шаге по времени промежуточных этапов, на каждом из которых записывается одномерная аппроксимация по одному из пространственных направлений. Многомерная задача «расщепляется» на последовательность одномерных задач по каждой из координат. Поэтому такие схемы называют *схемами расщепления по координатам*.

Заметим, что в подобных схемах отсутствует аппроксимация на каждом промежуточном этапе, т. е. на промежуточных этапах используемые одномерные разностные схемы не аппроксимируют исходное уравнение. Здесь имеет место лишь суммарная аппроксимация на слоях с целыми номерами. Погрешности аппроксимаций промежуточных слоев при суммировании уничтожаются. Такие схемы с суммарной аппроксимацией называются *аддитивными*.

Схема расщепления по координатам для двумерного уравнения теплопроводности может быть записана в виде

$$\frac{\tilde{u}_{ij} - u_{ij}^k}{\tau} = \frac{\tilde{u}_{i+1,\,j} - 2\tilde{u}_{ij} + \tilde{u}_{i-1,\,j}}{h_1^2},$$

$$\frac{u_{ij}^{k+1} - \tilde{u}_{ij}}{\tau} = \frac{u_{i,\,j+1}^{k+1} - 2u_{ij}^{k+1} + u_{i,\,j-1}^{k+1}}{h_2^2},$$

$$i = 1, 2, \ldots, I-1, \quad j = 1, 2, \ldots, J-1.$$

Она фактически представляет собой двукратную неявную схему для одномерного уравнения теплопроводности: на первом этапе находятся вспомогательные значения $\tilde{u}_{ij}$, на втором — искомые значения сеточной функции u_{ij}^{k+1}. Получающиеся системы уравнений имеют трехдиагональные матрицы и могут быть решены с помощью метода прогонки. Схема безусловно устойчива, она сходится со скоростью $O(h_1^2 + h_2^2 + \tau)$.

Из построения локально-одномерной схемы ясно, что она легко обобщается на случай произвольного числа переменных. При этом каждая новая переменная требует введения одного промежуточного этапа на каждом шаге по времени.

Другая группа методов расщепления основана на расщеплении задачи по физическим процессам. На каждом шаге по времени исходная сложная задача, описывающая некоторый физический процесс при наличии нескольких влияющих на него факторов, расщепляется на более простые задачи.

В настоящее время имеется несколько *схем расщепления по физическим процессам* в вычислительной аэродинамике. Например, при исследовании течений сжимаемого газа каждый шаг по времени можно проводить в два этапа. На первом из них определяется изменение параметров течения под влиянием только давления без учета процессов переноса. Второй этап состоит в пересчете полученных на первом шаге промежуточных результатов с учетом процессов переноса. Изложение вопросов, связанных с построением указанных схем, можно найти в специальной литературе.

4. Уравнение Лапласа. Многие стационарные физические задачи[1)] (исследования потенциальных течений жидкости, определение формы нагруженной мембраны, задачи теплопроводности и диффузии в стационарных случаях и др.) сводятся к решению *уравнения Пуассона* вида

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = F(x, y, z). \tag{8.88}$$

Если $F(x, y, z) = 0$, то уравнение (8.88) называется *уравнением Лапласа*. Для простоты будем рассматривать двумерное уравнение Лапласа

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0. \tag{8.89}$$

Решение этого уравнения будем искать для некоторой ограниченной области G изменения независимых переменных x, y. Границей области G является замкнутая линия L. Для полной формулировки краевой задачи кроме уравнения Лапласа нужно задать граничное условие на границе L. Примем его в виде

$$U(x, y)|_L = \varphi(x, y). \tag{8.90}$$

Задача, состоящая в решении уравнения Лапласа (или Пуассона) при заданных значениях искомой функции на границе расчетной области, называется *задачей Дирихле*.

Одним из способов решения стационарных эллиптических задач, в том числе и краевой задачи (8.89), (8.90), является их сведение к решению некоторой фиктивной нестационарной задачи (гиперболической или параболической), найденное решение которой при достаточно больших значениях времени t близко к решению исходной задачи. Такой способ решения называется *методом установления*.

Поскольку решение $U(x, y)$ уравнения (8.89) не зависит от времени, то можно в это уравнение добавить равный нулю (при точном решении)

[1)] То есть такие, в которых рассматриваются явления, неизменные с течением времени.

член $\partial U/\partial t$. Тогда уравнение (8.89) примет вид

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}. \tag{8.91}$$

Это — известное нам уравнение теплопроводности, для которого в пп. 2, 3 уже строились разностные схемы. Остается только задать начальное условие. Его можно принять практически в произвольном виде, согласованном с граничными условиями. Положим

$$U|_{t=0} = \psi(x, y). \tag{8.92}$$

Граничное условие (8.90) при этом остается стационарным, т. е. не зависящим от времени.

Процесс численного решения уравнения (8.91) с условиями (8.92), (8.90) состоит в переходе при $t \to \infty$ от произвольного значения (8.92) к искомому стационарному решению. Счет ведется до выхода решения на стационарный режим. Естественно, ограничиваются решением при некотором достаточно большом t, если искомые значения на двух последовательных слоях совпадают с заданной степенью точности.

Метод установления фактически представляет итерационный процесс решения задачи (8.91) с условиями (8.92), (8.90), причем на каждой итерации значения искомой функции получаются путем численного решения некоторой вспомогательной задачи. В теории разностных схем показано, что этот итерационный процесс сходится к решению исходной задачи, если такое стационарное решение существует.

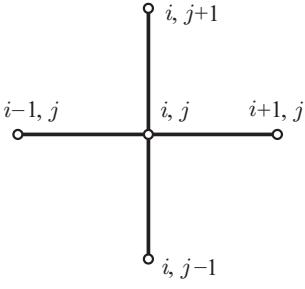


Рис. 8.27. Шаблон для уравнения Лапласа

Другой способ решения задачи Дирихле состоит в построении разностной схемы путем аппроксимации уравнения (8.89). Введем в прямоугольной области G сетку с помощью координатных прямых $x = \text{const}$ и $y = \text{const}$. Примем для простоты значения шагов по переменным x и y равными h (предполагается, что стороны области G соизмеримы). Значения функции U в узлах (x_i, y_j) заменим значениями сеточной функции u_{ij}. Тогда, аппроксимируя в уравнении (8.89) вторые производные с помощью отношений конечных разностей, получим разностное уравнение (шаблон изображен на рис. 8.27)

$$\frac{u_{i+1,\,j} - 2u_{ij} + u_{i-1,\,j}}{h^2} + \frac{u_{i,\,j+1} - 2u_{ij} + u_{i,\,j-1}}{h^2} = 0. \tag{8.93}$$

С помощью данного уравнения можно записать систему линейных алгебраических уравнений относительно значений сеточной функции

в узлах в виде

$$u_{i+1,\,j} + u_{i-1,\,j} + u_{i,\,j+1} + u_{i,\,j-1} - 4u_{ij} = 0, \quad i = 1, 2, \ldots, I-1, \quad j = 1, 2, \ldots, J-1. \tag{8.94}$$

Значения сеточной функции в узлах, расположенных на границе расчетной области, могут быть найдены из граничного условия (8.90):

$$u_{0j} = \varphi(x_0, y_j), \quad u_{Ij} = \varphi(x_I, y_j), \quad j = 0, 1, \ldots, J;$$
$$u_{i0} = \varphi(x_i, y_0), \quad u_{iJ} = \varphi(x_i, y_J), \quad i = 0, 1, \ldots, I.$$

В теории разностных схем доказывается, что решение построенной разностной задачи существует, а сама схема устойчива.

Перейдем теперь, к практическому вычислению искомых значений, т. е. к решению системы (8.94). Каждое уравнение системы (за исключением тех, которые соответствуют узлам, расположенным вблизи границ)

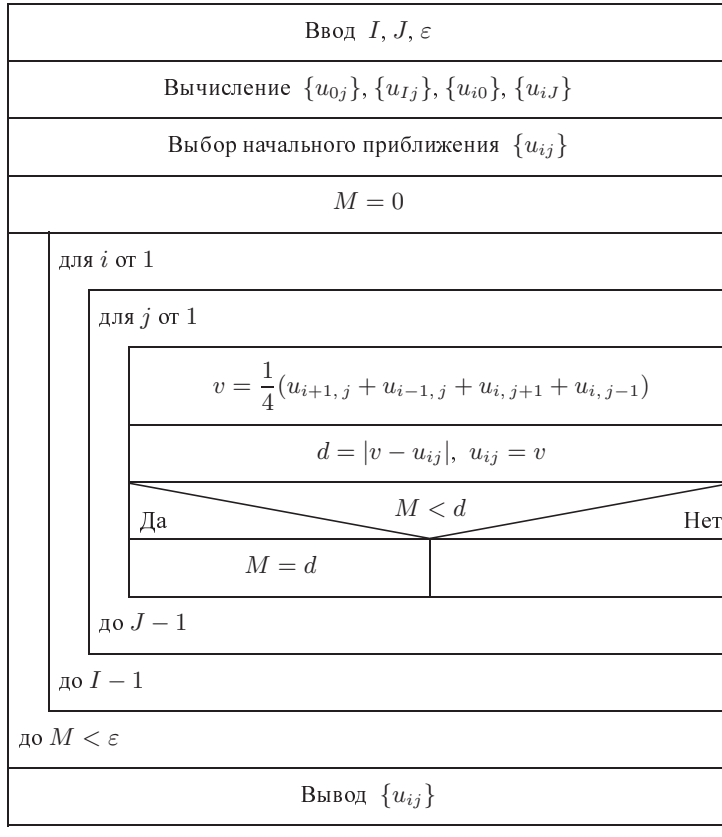


Рис. 8.28. Алгоритм решения задачи Дирихле

содержит пять неизвестных. Одним из наиболее распространенных методов решения этой системы линейных уравнений является итерационный метод. Каждое из уравнений записываем в виде, разрешенном относительно значения u_{ij} в центральном узле (см. рис. 8.27):

$$u_{ij} = \frac{1}{4}\,(u_{i+1,\,j} + u_{i-1,\,j} + u_{i,\,j+1} + u_{i,\,j-1}). \tag{8.95}$$

Алгоритм решения задачи Дирихле с использованием итерационного метода Гаусса – Зейделя решения системы разностных уравнений (8.95) изображен на рис. 8.28. В алгоритме предусмотрен выбор начальных значений u_{ij}. Иногда полагают $u_{ij} = 0$ для всех i, j. Итерационный процесс контролируется максимальным отклонением M значений сеточной функции в узлах для двух последовательных итераций. Если его величина достигнет некоторого заданного малого числа ε, итерации прекращаются и происходит вывод результатов.

Рассмотренные разностные схемы метода сеток используют конечно-разностные аппроксимации входящих в уравнения производных по всем переменным. В ряде случаев уравнение с частными производными удобно привести к системе обыкновенных дифференциальных уравнений, в которых оставлены производные искомой функции лишь по одной переменной.

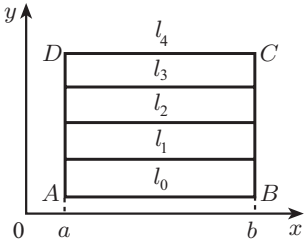


Рис. 8.29

Такой способ можно использовать и для решения уравнения Лапласа (8.89). Пусть требуется решить для него задачу Дирихле в прямоугольнике $ABCD$ (рис. 8.29). Разобьем прямоугольник на полосы с помощью прямых, параллельных оси x. Для определенности проведем три отрезка l_1, l_2, l_3, которые разделят прямоугольник на четыре полосы постоянной ширины h. Решение U задачи Дирихле приближенно заменим набором функций u_i, каждая из которых определена на отрезке l_i и зависит только от одной переменной x, т. е. $u_i = u_i(x)$ $(i = 1, 2, 3)$. На отрезках l_0 и l_4 значения $u_0(x)$ и $u_4(x)$ заданы граничными условиями.

Построим разностную схему для определения значений функций $u_i(x)$. Аппроксимируя в уравнении (8.89) вторую производную по y с помощью отношения конечных разностей, получаем

$$\frac{d^2u_i}{dx^2} + \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = 0, \quad i = 1, 2, 3. \tag{8.96}$$

Таким образом, решение задачи Дирихле (8.89), (8.90) сводятся к решению краевой задачи для системы обыкновенных дифференциальных уравнений (8.96) относительно значений искомой функции вдоль прямых l_1, l_2, l_3. В этом состоит *метод прямых*. Граничные условия для

уравнений (8.96) при $x = a,\ x = b$ можно получить из уравнений

$$u_i(a) = \varphi(a, y_i), \quad u_i(b) = \varphi(b, y_i), \quad i = 1, 2, 3.$$

Направление дискретизации y обычно легко выбрать в тех случаях, когда заранее известен характер поведения искомой функции; это направление должно соответствовать направлению наибольшей гладкости функции.

Метод прямых широко, используется для решения нестационарных задач. Например, если имеются две независимые переменные x, t, а искомый параметр является гладкой функцией переменной x, то дискретизация вводится по этой переменной. Тогда исходная задача заменяется задачей Коши для системы обыкновенных дифференциальных уравнений вида

$$d\mathbf{u}/dt = f(\mathbf{u}, t), \quad \mathbf{u} = \{u_0, u_1, \ldots, u_n\}.$$

Упражнения

1. Решение линейного уравнения переноса ищется в ограниченной области, заданной в полярной системе координат (r, φ): $r_0 \leqslant r \leqslant r_1$, $0 \leqslant \varphi \leqslant \pi/2$. Сформулировать математическую постановку задачи и построить разностные схемы ее решения: а) явную; б) неявную.

2. Записать алгоритм решения смешанной задачи для одномерного линейного уравнения переноса с использованием неявной разностной схемы.

3. Решить задачу

$$\frac{\partial U}{\partial t} + \frac{\partial U}{\partial x} = 0, \quad 0 \leqslant x \leqslant 1, \quad 0 \leqslant t \leqslant 1,$$
$$U(x, 0) = x^2, \quad U(0, t) = t^2:$$

а) аналитически;
б) вручную по явной схеме для одного временного слоя; принять $h = \tau = 0.1$;
в) с помощью компьютера; при этом убедиться, что явная схема условно устойчива, а неявная безусловно устойчива.

4. Модифицировать алгоритм решения двумерного уравнения переноса (см. рис. 8.13) для случая, когда число слоев K не является кратным L, и необходимо вывести результаты на последнем слое.

5*. Записать в укрупненном виде алгоритм нахождения разрывного решения квазилинейного уравнения переноса с использованием метода с выделением разрыва.

6. Записать алгоритм решения квазилинейного уравнения переноса по схеме сквозного счета с искусственной вязкостью.

7. Аппроксимировать граничное условие для незакрепленного конца стержня (8.72) со вторым порядком.

8. Как изменится алгоритм решения волнового уравнения (см. рис. 8.22), если требуется с целью экономии памяти машины хранить не все результаты, а лишь значения сеточной функции на трех последовательных слоях?

9. Записать алгоритм решения смешанной задачи для одномерного волнового уравнения по неявной схеме.

10. Решить задачу

$$\frac{\partial^2 U}{\partial t^2} = 4\frac{\partial^2 U}{\partial x^2}, \quad 0 \leqslant x \leqslant 1, \quad 0 \leqslant t \leqslant 1,$$
$$U(x,0) = x, \quad \partial U/\partial t(x,0) = 1, \quad U(0,t) = 0, \quad U(1,t) = e^{-t}:$$

а) вручную по явной схеме для одного временного слоя;
б) с помощью компьютера; при этом убедиться, что явная схема условно устойчива, а неявная — безусловно устойчива.

11. Записать алгоритм решения смешанной задачи для одномерного уравнения теплопроводности: а) с помощью явной схемы; б) с помощью неявной схемы.

12. Выполнить упр. 4 для уравнения $\partial U/\partial t = 2\partial^2 U/\partial x^2$. Какое из заданных условий при этом оказывается лишним?

13. Модифицировать алгоритм решения двумерного уравнения теплопроводности (см. рис. 8.26) так, чтобы результаты выдавались лишь на каждом пятом слое по времени.

14. Записать алгоритм решения задачи Дирихле методом установления.

15. Записать алгоритм решения смешанной задачи для двумерного уравнения теплопроводности по схеме переменных направлений.

16. Построить схему расщепления по координатам для: а) двумерного волнового уравнения; б) трехмерного уравнения теплопроводности.

17. Путем ручного счета получить численное решение на первой итерации по методу установления задачи

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0, \quad 0 \leqslant x \leqslant 1, \quad 0 \leqslant y \leqslant 1,$$
$$U(x,0) = U(1,0) = \sin \pi x, \quad U(0,y) = U(1,y) = \sin \pi y.$$

ГЛАВА 9

ИНТЕГРАЛЬНЫЕ УРАВНЕНИЯ

§ 4. Постановка задач

1. Вводные замечания. *Интегральным уравнением* называется такое уравнение, неизвестная функция в котором содержится под знаком интеграла. В общем случае интегральное уравнение имеет вид

$$\int_a^b K(x, s, y(s))\, ds = f(x, y(x)), \quad a \leqslant x \leqslant b. \tag{9.1}$$

Здесь x — независимая переменная, $y(x)$ — искомая функция, $K(x, s, y)$ — *ядро* интегрального уравнения, $f(x, y)$ — правая часть уравнения, s — переменная интегрирования.

К интегральным уравнениям приводят многие инженерные задачи (в радиотехнике, газовой динамике, квантовой механике и т. п.). Интегральная форма уравнений движения в виде законов сохранения используется также и при построении консервативных разностных схем для некоторых типов задач (в частности, в механике сплошной среды).

Для решения некоторых задач удобнее использовать интегральные уравнения, чем дифференциальные. Например, постановку задачи Коши

$$dy/dx = f(x, y), \quad y(x_0) = y_0$$

можно представить в виде интегрального уравнения

$$y(x) = y_0 + \int_{x_0}^{x} f(s, y(s))\, ds.$$

Таким образом, интегральное уравнение содержит полную постановку задачи, и дополнительные условия (начальные или граничные) для него задавать не нужно.

Отметим еще одно преимущество интегральных уравнений. Уравнение (9.1) записано для случая одной независимой переменной x. Однако легко записать его многомерный аналог при наличии независимых переменных $x_1, x_2, \ldots, x_n$. Для некоторой области G в рассматриваемом n-мерном пространстве многомерное интегральное уравнение можно записать в виде

$$\int_G K(x_1, \ldots, x_n, s_1, \ldots, s_n, y(s_1, \ldots, s_n))\, ds_1 \ldots ds_n = \\ = f(x_1, \ldots, x_n, y(x_1, \ldots, x_n)).$$

Методы решения одномерных уравнений естественно обобщаются на случай многомерных интегральных уравнений (одномерные интегралы заменяются многомерными). В то же время при рассмотрении дифференциальных уравнений переход от одномерного случая (обыкновенные уравнения) к многомерному (уравнения с частными производными) требует совершенно других подходов и методов решения.

2. Виды интегральных уравнений. Ограничимся рассмотрением одномерных уравнений (9.1). Приведем некоторые частные случаи таких уравнений, которые, с одной стороны, важны в практических приложениях и, с другой стороны, наиболее изучены.

Уравнения (9.1), в которые искомая функция входит линейно, называются *линейными интегральными уравнениями*. Одним из них является *уравнение Фредгольма первого рода*

$$\int_a^b K(x,s)y(s)\,ds = f(x), \quad a \leqslant x \leqslant b. \tag{9.2}$$

Уравнение Фредгольма второго рода имеет вид

$$y(x) - \lambda \int_a^b K(x,s)y(s)\,ds = f(x), \quad a \leqslant x \leqslant b. \tag{9.3}$$

В уравнениях Фредгольма ядро $K(x,s)$ определено и ограничено на квадрате $a \leqslant x \leqslant b$, $a \leqslant s \leqslant b$. Если $K(x,s) = 0$ при $s > x$, т. е. ядро отлично от нуля только на треугольнике $a \leqslant s \leqslant x$, $a \leqslant x \leqslant b$, то уравнения (9.2) и (9.3) переходят в *уравнения Вольтерра* соответственно *первого* и *второго рода*:

$$\int_a^x K(x,s)y(s)\,ds = f(x), \tag{9.4}$$

$$y(x) - \lambda \int_a^x K(x,s)y(s)\,ds = f(x). \tag{9.5}$$

Мы будем рассматривать задачи для уравнений второго рода. Задачи для уравнений первого рода являются некорректно поставленными. Их рассмотрение выходит за рамки данного краткого курса. Заметим лишь, что для решения некорректных задач, т. е. уравнений (9.2) или (9.4), могут быть использованы методы регуляризации.

Если правая часть уравнения (9.3) равна нулю, то получается *однородное уравнение Фредгольма второго рода*, которое можно записать в виде

$$y(x) = \lambda \int_a^b K(x,s)y(s)\,ds, \quad a \leqslant x \leqslant b. \tag{9.6}$$

Это уравнение допускает нулевое (тривиальное) решение $y(x)=0$. Для него может быть поставлена задача на собственные значения. Параметры λ_i, при которых уравнение (9.6) имеет отличные от нуля решения $y = \varphi_i(x)$, называются *собственными значениями* ядра $K(x,s)$ или уравнения (9.6), а отвечающие им решения $y = \varphi_i(x)$ — *собственными функциями*.

Теорема Фредгольма. *Если λ не является собственным значением ядра $K(x,s)$, то неоднородное уравнение* (9.3) *имеет единственное непрерывное решение $y(x)$ при $x\in[a,b]$, в противном случае данное неоднородное уравнение или не имеет решений, или имеет их бесчисленное множество.*

В практических приложениях важную роль играют уравнения Фредгольма второго рода с вещественным *симметричным ядром* $K(x,s)$, т. е. когда

$$K(x,s) = K(s,x).$$

Симметричное ядро обладает следующими свойствами:

1) симметричное ядро имеет хотя бы одно собственное значение;

2) все собственные значения симметричного ядра действительны;

3) собственные функции $\varphi_i(x)$ симметричного ядра ортогональны, т. е.

$$\int_a^b \varphi_i(x)\varphi_j(x) = 0, \quad i \neq j.$$

Уравнение Вольтерра (9.5) не имеет собственных значений. Соответствующее однородное уравнение, т. е. при $f(x) = 0$, имеет только тривиальное решение $y(x) = 0$. Следовательно, неоднородное уравнение (9.5) всегда при любом значении λ имеет решение, и при том единственное.

Итак, основными задачами для рассматриваемых интегральных уравнений являются:

1) нахождение решения неоднородного интегрального уравнения при заданном значении параметра λ;

2) вычисление собственных значений и отыскание соответствующих им собственных функций однородного интегрального уравнения.

§ 5. Методы решения

1. Методы последовательных приближений. Это простейшие методы решения интегральных уравнений, использовавшиеся еще задолго до появления компьютеров. Рассмотрим уравнение Фредгольма, записав его в виде

$$y(x) = f(x) + \lambda \int_a^b K(x,s)y(s)\,ds. \tag{9.7}$$

В дальнейшем под уравнением Фредгольма и Вольтерра будем подразумевать соответствующие уравнения второго рода.

Для решения уравнения (9.7) построим итерационный процесс, аналогичный методу простой итерации для нелинейного уравнения. Пусть $y_0(x)$ — начальное приближение искомой функции $y(x)$ (на практике обычно полагают $y_0(x) = 0$). Тогда, подставляя $y_0(x)$ в правую часть уравнения (9.7), получаем выражение для первого приближения:

$$y_1(x) = f(x) + \lambda \int_a^b K(x,s) y_0(s)\, ds.$$

Аналогично, подставляя найденное приближение в подынтегральное выражение, находим $y_2(x)$ и т. д. Для любого $(k+1)$-го приближения получим

$$y_{k+1}(x) = f(x) + \lambda \int_a^b K(x,s) y_k(s)\, ds, \quad k = 0, 1, \ldots \tag{9.8}$$

При достаточно малом значении $|\lambda|$ и ограниченном ядре $K(x,s)$, а именно, при

$$q = M|\lambda|(b-a) < 1, \quad M = \max_{x,s} |K(x,s)| \tag{9.9}$$

итерационный процесс (9.8) сходится равномерно по x, причем для погрешности $\varepsilon_k = |y_k(x) - y(x)|$ имеет место неравенство

$$\varepsilon_k \leqslant q^k \varepsilon_0, \quad k = 1, 2, \ldots \tag{9.10}$$

Сходимость итерационного процесса, при которой выполнено (9.10), называется *линейной* или *сходимостью со скоростью геометрической прогрессии*

Одним из вариантов метода последовательных приближений является метод, в котором используются степенные ряды. Он состоит в том, что искомое решение $y(x)$ разлагается в ряд по степеням λ:

$$y(x) = \sum_{k=0}^{\infty} \lambda^k g_k(x). \tag{9.11}$$

Подставляя это разложение в исходное уравнение (9.7) и приравнивая выражения при одинаковых степенях λ, получаем следующие рекуррентные соотношения:

$$g_0(x) = f(x), \quad g_k(x) = \lambda \int_a^b K(x,s) g_{k-1}(s)\, ds, \quad k = 1, 2, \ldots \tag{9.12}$$

При ограниченных $K(x,s)$ и $f(x)$ ряд (9.11) сходится, если выполняется условие (9.9).

Среди других приближенных методов отметим метод аппроксимации ядра данного интегрального уравнения вырожденным ядром. *Вырожденным ядром* уравнения Фредгольма называется ядро, которое может быть представлено в виде суммы конечного числа членов:

$$K(x,s)=\sum_{i=1}^{n}\psi_i(x)\chi_i(s),$$

т. е. каждый член разложения можно представить в виде произведения функций одной переменной $\psi_i(x)$ и $\chi_i(s)$. Вырожденное ядро имеет n собственных значений. С помощью такого ядра в ряде случаев удается аппроксимировать ядро данного уравнения, и решение полученного аппроксимирующего уравнения принимается в качестве приближенного решения исходного уравнения.

Для решения интегральных уравнений используется также *метод моментов*, основанный на использовании метода Галеркина. Здесь, как и при замене ядра вырожденным, для приближения решения строится аппроксимирующая система функций. Минимизация невязки аппроксимирующего уравнения проводится путем ее ортогонализации к базисным функциям.

В практических вычислениях рассмотренные методы сейчас используются сравнительно редко, поскольку присутствующие в аппроксимирующих выражениях (9.8) или (9.12) интегралы, как правило, не удается непосредственно вычислять в элементарных функциях. Однако эти методы полезны для нахождения первых приближений к решению.

2. Численные методы. Эти методы называют также *квадратурными*. Они основаны на использовании формул численного интегрирования для вычисления определенных интегралов, входящих в интегральные уравнения. Численные методы получили особенно широкое распространение в связи с внедрением компьютеров, хотя эти методы можно использовать и в ручном счете при небольшом числе узлов. Численные методы могут применяться для решения как линейных, так и нелинейных интегральных уравнений.

Рассмотрим нелинейное интегральное уравнение вида

$$\int_a^b K(x,s,y(s))\,ds=f(x,y(x)),\quad a\leqslant x\leqslant b. \tag{9.13}$$

Разобьем отрезок $[a,b]$ на части точками $x_i=a+ih$ $(i=0,1,\ldots,n)$. Заменим интеграл в уравнении (9.13) некоторой квадратурной формулой с помощью значений сеточной функции u_i в узлах:

$$\sum_{i=1}^{n}c_iK(x_j,x_i,u_i)=f(x_j,u_j),\quad j=1,2,\ldots,n, \tag{9.14}$$

где c_i — коэффициенты квадратурной формулы численного интегрирования.

Мы получили систему нелинейных алгебраических уравнений. Решая систему (9.14), получаем значения сеточной функции в выбранных узлах отрезка $[a, b]$. Для практического решения этой системы можно использовать рассмотренные ранее методы, например метод Ньютона (см. гл. 5, § 3).

Вопрос о сходимости сеточного решения u_i к значениям искомой функции $y(x_i)$ при $n \to \infty$ может быть рассмотрен лишь для конкретного вида интегрального уравнения. В общем случае сходимость численного метода исследовать трудно.

Рассмотрим линейные интегральные уравнения. Запишем сеточное выражение (9.14) для однородного уравнения Фредгольма:

$$u_j = \lambda \sum_{i=1}^{n} c_i K(x_j, x_i) u_i,$$

или

$$\sum_{i=1}^{n} c_i K(x_j, x_i) u_i = \frac{1}{\lambda} u_j, \quad j = 1, 2, \ldots, n. \tag{9.15}$$

Система линейных уравнений в таком виде описывает задачу на собственные значения матрицы A, элементами которой являются числа $a_{ji} = c_i K(x_j, x_i)$ (см. гл. 4, § 4). Матрица A имеет n собственных значений (с учетом кратности), которые являются приближениями к собственным значениям однородного уравнения Фредгольма.

В случае неоднородного уравнения Фредгольма вместо однородной системы (9.15) получим следующую систему линейных алгебраических уравнений:

$$u_j - \lambda \sum_{i=1}^{n} c_i K(x_j, x_i) u_i = f(x_j), \quad j = 1, 2, \ldots, n. \tag{9.16}$$

Эта система уравнений может быть решена одним из рассмотренных ранее методов (см. гл. 4), например методом Гаусса. В соответствии с теоремой Фредгольма (см. § 1, п. 2) параметр λ не должен быть равен ни одному из собственных значений. Если он попадает в окрестность некоторого собственного значения, то система (9.16) становится плохо обусловленной, и сеточное решение u_i может сильно отличаться от искомых значений $y(x_i)$.

На практике обычно собственные значения интегрального уравнения неизвестны, поэтому ограничиваются исследованием практической сходимости. Оно состоит в проведении серии расчетов со сгущающейся сеткой. Если при этом наблюдается сходимость сеточных значений, то в качестве искомого решения принимаются результаты последнего расчета на густой сетке. При решении уравнения Вольтерра система линейных алгебраических уравнений (9.16) имеет треугольный вид, и она легко решается последовательным нахождением значений u_i (по аналогии с обратным ходом метода Гаусса).

Рассмотренный подход можно использовать и для решения многомерных интегральных уравнений. При этом в многомерной расчетной области значительно возрастает число узлов. Для решения таких задач требуется большой объем памяти компьютера; системы уравнений в этих случаях более целесообразно решать итерационными методами.

Пример. Пусть задано уравнение

$$y(x) - \lambda \int_0^1 e^{-(x+s)} y(s)\, ds = x. \tag{9.17}$$

Используя рассмотренные выше методы, нужно найти значения искомой функции $y(x)$ на отрезке $[0, 1]$.

Решение. Для применения итерационного процесса (9.8) для приближенного решения данного интегрального уравнения примем в качестве нулевого приближения $y_0(x) = 0$. Тогда

$$y_1(x) = x + \lambda \int_0^1 e^{-(x+s)} y_0(s)\, ds = x.$$

Подставляя полученное приближение $y_k(s) = s$ $(k = 1)$ в (9.8) и используя формулу интегрирования по частям

$$\int_0^1 u\, dv = uv\Big|_0^1 - \int_0^1 v\, du,$$

получаем следующее приближение к решению:

$$y_2(x) = x + \lambda \int_0^1 s e^{-(x+s)} ds = x - \lambda s e^{-(x+s)}\Big|_0^1 + \lambda \int_0^1 e^{-(x+s)} ds =$$

$$= x - \lambda e^{-(x+1)} - \lambda e^{-(x+s)}\Big|_0^1 = x - 2\lambda e^{-(x+1)} + \lambda e^{-x}.$$

Аналогично находим

$$y_3(x) = x + \lambda \int_0^1 e^{-(x+s)} y_2(s)\, ds = x + \lambda e^{-x}(a_1 + a_2 \lambda), \tag{9.18}$$

$$a_1 = 1 - 2e^{-1}, \quad a_2 = \frac{1 - 2e^{-1} - e^{-2} + 2e^{-3}}{2}.$$

В данном случае можно построить любое приближение к решению уравнения (9.17). Сходимость построенного итерационного процесса оценивается с помощью условия (9.9), которое дает ограничение на параметр λ:

$$|\lambda| < \frac{1}{M(b-a)}. \tag{9.19}$$

Для рассматриваемого примера имеем

$$b-a=1, \quad M = \max_{x,s} |K(x,s)| = \max_{x,s} e^{-(x+s)} = 1.$$

Следовательно, из (9.19) получаем условие $|\lambda| < 1$.

Если для решения уравнения (9.17) использовать метод степенных рядов, то искомую функцию нужно представить в виде (9.11), а из рекуррентных соотношений (9.12) найти члены разложения:

$$g_0(x) = f(x) = x,$$

$$g_1(x) = \int_0^1 K(x,s) g_0(s)\, ds = \int_0^1 s e^{-(x+s)} ds = e^{-x} - 2e^{-(x+1)},$$

$$g_2(x) = \int_0^1 K(x,s) g_1(s)\, ds = \int_0^1 e^{-(x+s)} \left[e^{-s} - 2e^{-(s+1)}\right] ds =$$

$$= \frac{1}{2} e^{-x} - e^{-(x+1)} - \frac{1}{2} e^{-(x+2)} + e^{-(x+2)},$$

. . . . . .

Подставляя вычисляемые значения $g_i(x)$ в выражение (9.11), находим приближение для искомой функции $y(x)$. Если ограничиться тремя членами ряда (9.11), то результаты совпадают с полученным ранее приближением (9.18). При $|\lambda| < 1$ ряд (9.11) сходится к искомому решению.

§ 6. Сингулярные уравнения

1. Сингулярные интегралы. Рассмотренные выше интегральные уравнения содержали неособые интегралы, подынтегральная функция которых определена и непрерывна на отрезке интегрирования. Однако при решении ряда практических задач приходится сталкиваться с уравнениями, содержащими несобственные интегралы. Рассмотрим некоторые виды интегралов, имеющих непосредственное отношение к решению практически важных интегральных уравнений. Эти интегралы представляют также и самостоятельный интерес.

Пусть подынтегральная функция $f(x)$ интеграла

$$\int_a^b f(x)\, dx \tag{9.20}$$

обращается в некоторой точке c отрезка $[a,b]$ в бесконечность, т. е. интеграл несобственный. Тогда его можно попытаться вычислить следующим образом:

$$\int_a^b f(x)\,dx = \lim_{\varepsilon_1\to 0}\int_a^{c-\varepsilon_1} f(x)\,dx + \lim_{\varepsilon_2\to 0}\int_{c+\varepsilon_2}^b f(x)\,dx. \tag{9.21}$$

Здесь ε_1, ε_2 — некоторые положительные числа, которые стремятся к нулю независимо друг от друга. Если выражения в правой части (9.21) существуют, то несобственный интеграл (9.20) сходится.

При решении ряда прикладных задач встречаются несобственные интегралы вида

$$\int_a^b \frac{dx}{x-c}, \quad c\in[a,b]. \tag{9.22}$$

В соответствии с (9.21) можно записать

$$\int_a^b \frac{dx}{x-c} = \lim_{\varepsilon_1\to 0}\int_a^{c-\varepsilon_1}\frac{dx}{x-c} + \lim_{\varepsilon_2\to 0}\int_{c+\varepsilon_2}^b \frac{dx}{x-c} = \lim_{\varepsilon_1\to 0}\ln\frac{\varepsilon_1}{c-a} + \lim_{\varepsilon_2\to 0}\ln\frac{b-c}{\varepsilon_2}.$$

Поскольку оба предела равны бесконечности, то интеграл (9.22) здесь является расходящимся.

Однако этот интеграл можно понимать и в другом смысле, полагая

$$\varepsilon_1 = \varepsilon_2 = \varepsilon.$$

В этом случае

$$\int_a^b \frac{dx}{x-c} = \lim_{\varepsilon\to 0}\left(\int_a^{c-\varepsilon}\frac{dx}{x-c} + \int_{c+\varepsilon}^b \frac{dx}{x-c}\right) =$$

$$= \lim_{\varepsilon\to 0}\left(\ln\frac{\varepsilon}{c-a} + \ln\frac{b-c}{\varepsilon}\right) = \lim_{\varepsilon\to 0}\ln\frac{b-c}{c-a} = \ln\frac{b-c}{c-a}.$$

Интеграл в таком смысле называется *интегралом в смысле главного значения по Коши* или *сингулярным интегралом*.

Аналогично можно ввести интегралы более общего вида

$$\int_a^b \frac{\gamma(x)\,dx}{x-c} = \lim_{\varepsilon\to 0}\left(\int_a^{c-\varepsilon}\frac{\gamma(x)\,dx}{x-c} + \int_{c+\varepsilon}^b \frac{\gamma(x)\,dx}{x-c}\right).$$

Оказывается, что в таком смысле интеграл существует при любой функции $\gamma(x)$, которую можно представить в виде

$$\gamma(x) = \frac{\varphi(x)}{(x-a)^\nu(b-x)^\mu}, \quad \nu<1, \quad \mu<1.$$

Здесь функция $\varphi(x)$ удовлетворяет некоторому условию, называемому *условием Гёльдера степени α на отрезке $[a,b]$*, которое состоит в том, что для любых двух точек x_1, x_2 этого отрезка

$$|\varphi(x_1)-\varphi(x_2)|\leqslant A|x_1-x_2|^{\alpha},$$
$$0<\alpha\leqslant 1,\quad A=\text{const}.$$

В этом случае говорят, что *функция $\varphi(x)$ принадлежит классу $H(\alpha)$*: $\varphi(x)\in H(\alpha)$. В частности, функция, имеющая на отрезке $[a,b]$ ограниченную производную, принадлежит классу $H(\alpha)$ с любым $0<\alpha\leqslant 1$.

В ряде приложений встречаются также интегралы вида

$$\int\limits_0^{2\pi}\gamma(\theta)\operatorname{ctg}\frac{\theta-\beta}{2}\,d\theta,\quad b\in[0,2\pi],\quad \gamma(\theta)\in H(\alpha). \tag{9.23}$$

Такие интегралы называют *интегралами с ядром Гильберта*. Они существуют в рассмотренном выше смысле, т. е. как сингулярные.

Для сингулярных интегралов, как и в случае определенных интегралов, справедлива формула замены переменной $x=x(t)$, однако производная $x=x'(t)$ должна принадлежать классу H в окрестности точки t_0 такой, что $c=x(t_0)$.

Пример. Докажем, что при любом значении $c\in[-1,1]$ имеет место тождество

$$\int\limits_{-1}^{1}\sqrt{\frac{1-x}{1+x}}\,\frac{dx}{c-x}=\pi.$$

Для доказательства сделаем в левой части дробно-линейную подстановку $t=\sqrt{(1-x)/(1+x)}$. В этом случае, как легко убедиться, функция $x(t)=(1-t^2)/(1+t^2)$ имеет ограниченную производную на всей числовой прямой и, следовательно, принадлежит классу H. Имеем:

$$\int\limits_{-1}^{1}\sqrt{\frac{1-x}{1+x}}\,\frac{dx}{c-x}=\int\limits_0^{+\infty}\frac{t^2\,dt}{(1+t^2)[t^2(1+c)-(1-c)]}=$$
$$=2\operatorname{arctg}t\Big|_0^{+\infty}+\sqrt{\frac{1-c}{1+c}}\,\ln\left|\frac{t\sqrt{1+c}-\sqrt{1-c}}{t\sqrt{1+c}+\sqrt{1-c}}\right|\Bigg|_0^{+\infty}=\pi.$$

Рассмотрим вопросы, связанные с построением методов численного интегрирования для рассматриваемых особых случаев. Оказывается, что исходя из самого определения сингулярного интеграла (вырезается симметричная окрестность точки, в которой он вычисляется), можно построить простую формулу типа прямоугольников для вычисления сингулярных интегралов.

Пусть надо вычислить сингулярный интеграл на отрезке $[-1,1]$ в точке c. Возьмем равноотстоящие на шаг h точки x_1, x_2, ..., x_n такие, что

точка c делит пополам отрезок между ближайшими к ней точками из этого семейства. При этом крайние точки x_1 и x_n лежат на расстоянии не менее полушага от концов отрезка. Тогда

$$\int_{-1}^{1} \frac{\gamma(x)\,dx}{x-c} = \sum_{k=1}^{n} \frac{\gamma(x_k)h}{x_k - c}\,.$$

Разность между точным значением интеграла и значением полученной квадратурной суммы есть величина порядка $\ln n/n^{\alpha}$, если $\varphi(x) \in H(\alpha)$.

В приложениях, как правило, такие интегралы надо вычислять сразу в большом количестве точек, равномерно расположенных на отрезке $[-1, 1]$. Поэтому выбирают два семейства точек:

$$x_k = -1 + kh, \quad h = \frac{2}{n+1}, \quad k = 0, 2, \ldots, n,$$

$$c_k = x_k + \frac{h}{2}, \quad k = 0, 1, \ldots, n,$$

и пользуются формулой

$$\int_{-1}^{1} \frac{\gamma(x)\,dx}{x-c_m} = \sum_{k=1}^{n} \frac{\gamma(x_k)h}{x_k - c_m}, \quad m = 0, 1, \ldots, n.$$

Для интеграла с ядром Гильберта (9.23) используют следующую квадратурную формулу. Возьмем два семейства точек:

$$\theta_k = k\Big(\frac{2\pi}{n}\Big), \quad \beta_k = \theta_k + \frac{\pi}{n}, \quad k = 0, 1, \ldots, n-1,$$

Тогда

$$\int_{0}^{2\pi} \gamma(\theta) \operatorname{ctg} \frac{\theta - \beta_m}{2}\, d\theta \approx \sum_{k=0}^{n-1} \frac{2\pi}{n} \gamma(\theta_k) \operatorname{ctg} \frac{\theta_k - \beta_m}{2}\,,$$

$$m = 0, 1, \ldots, n-1.$$

Если функция $\gamma(\theta)$ принадлежит $H(\alpha)$ на отрезке $[0, 2\pi]$ и периодическая, то разность интеграла и суммы для любого m есть величина порядка $\ln n/n^{\alpha}$. Если же n нечетно и $\gamma^{(r)}(\theta) \in H(\alpha)$, то эта разность будет величиной порядка $\ln n/n^{r+\alpha}$.

Для интеграла на отрезке в частных случаях можно также указать простые *квадратурные формулы типа Гаусса*, дающие хорошую сходимость:

$$\int_{-1}^{1} \frac{\gamma(x)\,dx}{x-c_m} = \sum_{k=1}^{n} \frac{a_k \varphi(x_k)h}{x_k - c_m}\,, \quad m = 1, 2, \ldots, n-1,$$

$$\gamma(x) = \frac{\varphi(x)}{\sqrt{1-x^2}}\,, \quad \varphi^{(r)}(\theta) \in H(\alpha),$$

$$x_k = \cos \frac{2k-1}{2n}\,\pi, \quad k = 1, 2, \ldots, n,$$

$$c_m = \cos\frac{m\pi}{n}, \quad m = 1, 2, \ldots, n-1,$$
$$a_k = \frac{\pi}{n}, \quad k = 1, 2, \ldots, n.$$

Если

$$\gamma(x) = \sqrt{1-x^2}\varphi(x),$$

то

$$x_k = \cos\frac{k}{n+1}\,\pi, \quad k = 1, 2, \ldots, n,$$
$$c_m = \cos\frac{2m-1}{2(n+1)}\,\pi, \quad m = 1, 2, \ldots, n+1,$$
$$a_k = \frac{\pi}{n+1}\sin^2\frac{k}{n+1}\,\pi, \quad k = 1, 2, \ldots, n.$$

Если

$$\gamma(x) = \sqrt{\frac{1-x}{1+x}}\,\varphi(x),$$

то

$$x_k = \cos\frac{2k}{2n+1}\,\pi, \quad k = 1, 2, \ldots, n,$$
$$c_m = \cos\frac{2m-1}{2n+1}\,\pi, \quad m = 1, 2, \ldots, n,$$
$$a_k = \frac{4\pi}{2n+1}\sin^2\frac{k}{2n+1}\,\pi, \quad k = 1, 2, \ldots, n.$$

2. Численное решение сингулярных интегральных уравнений. Рассмотрим следующие сингулярные интегральные уравнения первого рода:

$$\frac{1}{\pi}\int\limits_{-1}^{1}\frac{\gamma(x)\,dx}{x-c} + \int\limits_{-1}^{1}K(c,x)\gamma(x)\,dx = f(c), \tag{9.24}$$

$$\frac{1}{2\pi}\int\limits_{0}^{2\pi}\gamma(\theta)\operatorname{ctg}\frac{\theta-\beta}{2}\,d\theta + \int\limits_{0}^{2\pi}K(\beta,\theta)\gamma(\theta)\,d\theta = f(\beta). \tag{9.25}$$

Здесь функции $K(\beta,\theta)$, $f(\beta)$ принадлежат классу $H(\alpha)$ соответственно на отрезках $[-1,1]$ и $[0,2\pi]$, причем они периодические по обеим переменным с периодом 2π.

Решение уравнения (9.24) не единственно. Это уравнение может иметь три типа решений, называемых *решениями индекса* $\varkappa$ ($\varkappa = 1, 0, -1$). Они имеют вид

$$\gamma_\varkappa(x) = \omega_\varkappa(x)\varphi(x),$$
$$\omega_1(x) = \frac{1}{\sqrt{1-x^2}}, \quad \omega_0(x) = \sqrt{\frac{1-x}{1+x}}, \quad \omega_{-1}(x) = \sqrt{1-x^2}.$$

Функция $\varphi(x)$ принадлежит классу H на отрезке $[-1,1]$. Функцию $\omega_\varkappa(x)$ называют *весовой функцией* решения данного индекса. Для нулевого индекса весовая функция может иметь вид

$$\omega_0(x)=\sqrt{\frac{1+x}{1-x}}\,.$$

Если в уравнении (9.24) $K(c,x)=0$, то оно называется *характеристическим*. Его решения даются формулой

$$\gamma_\varkappa(x)=-\frac{1}{\pi}\,\omega_\varkappa(x)\left[\int\limits_{-1}^{1}\frac{f(c)}{\omega_\varkappa(c)}\,\frac{dc}{c-x}-\nu_\varkappa C\right],$$

где $\nu_1=1$, $\nu_0=\nu_{-1}=0$, C — произвольная постоянная.

При $\varkappa=1$ единственное решение выделяется заданием значения интеграла

$$\int\limits_{-1}^{1}\gamma_1(x)\,dx=C.$$

При $\varkappa=-1$ функция $\gamma_{-1}(x)$ является решением характеристического уравнения только при условии

$$\int\limits_{-1}^{1}\frac{f(x)\,dx}{\sqrt{1-x^2}}=0.$$

В силу этого предполагают, что исходное уравнение (9.24) имеет единственное решение индекса 1 при заданном значении интеграла от решения, единственное решение индекса 0 и единственное решение индекса -1 при условии

$$\int\limits_{-1}^{1}\left[f(c)-\int\limits_{-1}^{1}K(c,x)\gamma(x)\,dx\right]\frac{dc}{\sqrt{1-c^2}}=0. \tag{9.26}$$

Для решения рассматриваемых сингулярных интегральных уравнений существует *метод дискретных особенностей*, основанный на приведенных выше квадратурных формулах. Он сводит задачу к решению систем линейных алгебраических уравнений. Приведем эти системы для случая равномерного расположения точек.

Для $\varkappa=1$ получается следующая система линейных алгебраических уравнений:

$$\frac{1}{\pi}\sum_{k=1}^{n}\frac{\gamma_n(x_k)h}{x_k-c_m}+\sum_{k=1}^{n}K(c_m,x_k)\gamma_n(x_k)h=f(c_m),$$

$$m=1,2,\ldots,n-1,$$

$$\sum_{k=1}^{n}\gamma_n(x_k)h=C;$$

для $\varkappa=0$

$$\frac{1}{\pi}\sum_{k=1}^{n}\frac{\gamma_n(x_k)h}{x_k-c_m}+\sum_{k=1}^{n}K(c_m,x_k)\gamma_n(x_k)h=f(c_m),$$
$$m=1,2,\ldots,n;$$

для $\varkappa=-1$

$$\gamma_{0n}+\frac{1}{\pi}\sum_{k=1}^{n}\frac{\gamma_n(x_k)h}{x_k-c_m}+\sum_{k=1}^{n}K(c_m,x_k)\gamma_n(x_k)h=f(c_m),$$
$$m=0,1,\ldots,n.$$

В последней системе γ_{0n} называется *регуляризирующей переменной*, причем $\gamma_{0n}\to 0$ при $n\to\infty$ тогда и только тогда, когда выполняется условие (9.26). Таким образом, величина γ_{0n} в расчете является индикатором его правильности.

Если использовать неравномерное разбиение, то системы линейных алгебраических уравнений примут вид:

для $\varkappa=1$

$$\frac{1}{\pi}\sum_{k=1}^{n}\frac{a_k\varphi_n(x_k)}{x_k-c_m}+\sum_{k=1}^{n}a_kK(c_m,x_k)\varphi_n(x_k)=f(c_m),$$
$$m=1,2,\ldots,n-1,$$
$$\sum_{k=1}^{n}a_k\varphi_n(x_k)=C,$$
$$a_k=\frac{\pi}{n},\quad x_k=\cos\frac{2k-1}{2n}\,\pi,\quad c_m=\cos\frac{m\pi}{n}\,;$$

для $\varkappa=0$

$$\frac{1}{\pi}\sum_{k=1}^{n}\frac{a_k\varphi_n(x_k)}{x_k-c_m}+\sum_{k=1}^{n}a_kK(c_m,x_k)\varphi_n(x_k)=f(c_m),$$
$$m=1,2,\ldots,n,$$
$$a_k=\frac{4\pi}{2n+1}\sin^2\frac{k}{2n+1}\,\pi,\quad x_k=\cos\frac{2k}{2n+1}\,\pi,\quad c_m=\cos\frac{2m-1}{2n+1}\,\pi;$$

для $\varkappa=-1$

$$\gamma_{0n}+\frac{1}{\pi}\sum_{k=1}^{n}\frac{a_k\varphi_n(x_k)}{x_k-c_m}+\sum_{k=1}^{n}a_kK(c_m,x_k)\varphi_n(x_k)=f(c_m),$$
$$m=1,2,\ldots,n+1,$$
$$a_k=\frac{\pi}{n+1}\sin^2\frac{k}{n+1}\,\pi,\quad x_k=\cos\frac{k}{n+1}\,\pi,\quad c_m=\cos\frac{2m-1}{2(n+1)}\,\pi.$$

Для характеристического уравнения (9.25) с ядром Гильберта при условии

$$\int_0^{2\pi} f(\beta)\,d\beta = 0$$

решение дается формулой

$$\gamma(\theta) = -\frac{1}{2\pi}\int_0^{2\pi} f(\beta)\operatorname{ctg}\frac{\beta-\theta}{2}\,d\beta + C,$$

где

$$\frac{1}{2\pi}\int_0^{2\pi}\gamma(\theta)\,d\theta = C.$$

Задание значения интеграла выделяет единственное решение. Поэтому будем предполагать, что уравнение с ядром Гильберта при известном значении интеграла имеет единственное решение. Для численного решения получается следующая система линейных алгебраических уравнений:

$$\gamma_{0n} + \frac{1}{2n+1}\sum_{k=0}^{2n}\gamma_n(\theta_k)\operatorname{ctg}\frac{\theta_k-\beta_m}{2} + \\ + \frac{2\pi}{2n+1}\sum_{k=0}^{2n}K(\beta_m,\theta_k)\gamma_n(\theta_k) = f(\beta_m),$$

$$m = 0, 1, \ldots, 2n,$$

$$\frac{1}{2n+1}\sum_{k=0}^{2n}\gamma_n(\theta_k) = C.$$

Приведенные системы линейных алгебраических уравнений метода дискретных особенностей могут быть использованы для вычисления значений $\gamma(x_k)$, $\varphi(x_k)$, $\gamma(\theta_k)$ в расчетных точках, которые аппроксимируют значения искомых функций $\gamma(x)$, $\varphi(x)$, $\gamma(\theta)$, описываемых сингулярными интегральными уравнениями (9.24), (9.25).

Упражнения

1. Свести к интегральному уравнению задачу из упр. 3 к гл. 7.

2. Записать в укрупненном виде алгоритм решения уравнения (9.7) методом последовательных приближений.

3*. Доказать линейную сходимость итерационного процесса (9.8) при выполнении условия (9.9).

4. Показать, что функция $x(t)$ из примера, приведенного в п. 1 § 3, принадлежит классу H.

5. Записать алгоритм решения уравнения (9.24) методом дискретных особенностей для случая равномерного разбиения отрезка. Алгоритм решения системы линейных алгебраических уравнений при этом не детализировать.

ПРИЛОЖЕНИЕ А

СТРУКТУРОГРАММЫ

Удобным средством графического представления алгоритмов являются структурограммы (*диаграммы Насси – Шнайдермана*). На таких диаграммах можно наглядно показать структуру алгоритма (отсюда и название). Алгоритм, представленный с помощью структурограммы, легко запрограммировать с соблюдением принципов *структурного программирования*. В частности, структурограммы (в отличие, например, от блок-схем) не допускают произвольную передачу управления (безусловный переход), которая затрудняет написание надежных и легко читаемых программ.

Структурограмма состоит из набора прямоугольных элементов. Каждый элемент соответствует либо некоторому действию, не требующему дальнейшей детализации, либо одной из алгоритмических конструкций (например, следованию, ветвлению или циклу).

Прямоугольники, расположенные друг под другом, обозначают *следование*, т. е. действия, выполняемые последовательно (см. рис А.1).

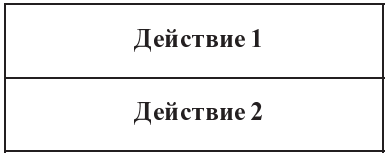


Рис. А.1. Действия, выполняемые последовательно

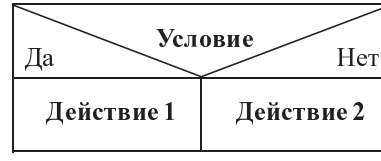


Рис. А.2. Условная конструкция

Обозначение *условной конструкции* (ветвления) показано на рис. А.2. Если **Условие** выполнено, выполняется **Действие 1**, если не выполнено — **Действие 2**. На рис. А.3, А.4 и А.5 показаны обозначения трех видов цикли-

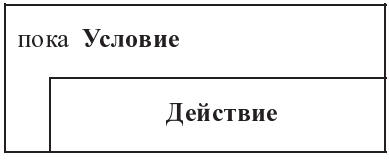


Рис. А.3. Цикл с предусловием

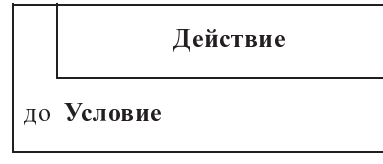


Рис. А.4. Цикл с постусловием

ческих конструкций: цикла с предусловием, цикла с постусловием и цикла с параметром (со счетчиком).

В цикле с *предусловием* предполагается, что **Условие** проверяется *перед* каждой *итерацией* цикла (выполнением действия). Если **Условие** *не выполнено*, цикл завершается.

В цикле с *постусловием* предполагается, что **Условие** проверяется *после* каждой итерации. Если **Условие** *выполнено*, цикл завершается.

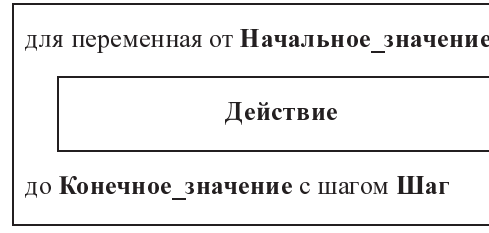


Рис. А.5. Цикл с параметром

В цикле с *параметром* значение переменной (параметра) цикла изменяется от **Начального значения** на первой итерации до **Конечного значения** на последней итерации с шагом **Шаг**. Если **Шаг** не указан, он полагается равным 1.

Рассмотренные здесь условные и циклические конструкции реализованы (иногда с некоторыми отличиями) в современных языках программирования высокого уровня.

Существуют обозначения и других алгоритмических конструкций — многовариантного ветвления, вызова процедуры, которые здесь рассматривать не будем.

Сравнить представление алгоритма в виде структурограммы и блок-схемы можно по рис. 1.1 и 1.2.

ПРИЛОЖЕНИЕ Б

МНОГОЧЛЕНЫ ЧЕБЫШЕВА

1. Многочлены Чебышева $T_n(x)$ (называемые также многочленами Чебышева *первого рода*):

$$T_n(x) = \frac{1}{2}\left[(x+\sqrt{1-x^2})^n + (x-\sqrt{1-x^2})^n\right] = \cos(n\arccos x),$$
$$n = 0, 1, \ldots,$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \ldots,$$

$$T_0(x) = 1,$$
$$T_1(x) = x,$$
$$T_2(x) = 2x^2 - 1,$$
$$T_3(x) = 4x^3 - 3x,$$
$$T_4(x) = 8x^4 - 8x^2 + 1,$$
$$T_5(x) = 16x^5 - 20x^3 + 5x,$$
$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1,$$
$$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x,$$
$$T_8(x) = 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1,$$
$$T_9(x) = 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x,$$
$$T_{10}(x) = 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1,$$
$$T_{11}(x) = 1024x^{11} - 2816x^9 + 2816x^7 - 1232x^5 + 220x^3 - 11x.$$

2. Представление степеней x через многочлены $T_n(x)$:

$$x^0 = 1 = T_0,$$
$$x^1 = T_1,$$
$$x^2 = \frac{1}{2}(T_0 + T_2),$$
$$x^3 = \frac{1}{4}(3T_1 + T_3),$$
$$x^4 = \frac{1}{8}(3T_0 + 4T_2 + T_4),$$
$$x^5 = \frac{1}{16}(10T_1 + 5T_3 + T_5),$$

$$x^6 = \frac{1}{32}\,(10T_0 + 15T_2 + 6T_4 + T_6),$$
$$x^7 = \frac{1}{64}\,(35T_1 + 21T_3 + 7T_5 + T_7),$$
$$x^8 = \frac{1}{128}\,(35T_0 + 56T_2 + 28T_4 + 8T_6 + T_8),$$
$$x^9 = \frac{1}{256}\,(126T_1 + 84T_3 + 36T_5 + 9T_7 + T_9),$$
$$x^{10} = \frac{1}{512}\,(126T_0 + 210T_2 + 120T_4 + 45T_6 + 10T_8 + T_{10}),$$
$$x^{11} = \frac{1}{1024}\,(462T_1 + 330T_3 + 165T_5 + 55T_7 + 11T_9 + T_{11}).$$

3. Выражение x^n через более низкие степени:

$$x^0 = T_1,$$
$$x^2 = \frac{1}{2}\,(1 + T_2),$$
$$x^3 = \frac{1}{4}\,(3x + T_3),$$
$$x^4 = \frac{1}{8}\,(8x^2 - 1 + T_4),$$
$$x^5 = \frac{1}{16}\,(20x^3 - 5x + T_5),$$
$$x^6 = \frac{1}{32}\,(48x^4 - 18x^2 + 1 + T_6),$$
$$x^7 = \frac{1}{64}\,(112x^5 - 56x^3 + 7x + T_7),$$
$$x^8 = \frac{1}{128}\,(256x^6 - 160x^4 + 32x^2 - 1 + T_8),$$
$$x^9 = \frac{1}{256}\,(576x^7 - 432x^5 + 120x^3 - 9x + T_9),$$
$$x^{10} = \frac{1}{512}\,(1280x^8 - 1120x^6 + 400x^4 - 50x^2 + 1 + T_{10}),$$
$$x^{11} = \frac{1}{1024}\,(2816x^9 - 2816x^7 + 1232x^5 - 220x^3 + 11x + T_{11}).$$

СПИСОК ЛИТЕРАТУРЫ

1. *Аоки М.* Введение в методы оптимизации: Основы и приложения нелинейного программирования / Пер. с англ. — М.: Наука, 1977.
2. *Бахвалов Н. С., Жидков Н. П., Кобельков Г. М.* Численные методы. — М.: Наука, 1987.
3. *Белоцерковский О. М.* Численное моделирование в механике сплошных сред. — М.: Физматлит, 1994.
4. *Белоцерковский О. М., Давыдов Ю. М.* Метод крупных частиц в газовой динамике. — М.: Наука, 1982.
5. *Белоцерковский С. М., Лифанов И. К.* Численные методы в сингулярных интегральных уравнениях. — М.: Наука, 1985.
6. *Березин И. С., Жидков Н. П.* Методы вычислений. Т. 1. — М.: Наука, 1966; Т. 2. — М.: Физматгиз, 1962.
7. *Васильев Ф. П.* Численные методы решения экстремальных задач. — М.: Наука, 1988.
8. *Воеводин В. В.* Вычислительные основы линейной алгебры. — М.: Наука, 1977.
9. *Волков Е. А.* Численные методы. — М.: Наука, 1987.
10. *Годунов С. К., Забродин А. В., Иванов М. Я., Крайко А. Н., Прокопов Г. П.* Численное решение многомерных задач газовой динамики. — М.: Наука, 1976.
11. *Годунов С. К., Рябенький В. С.* Разностные схемы. — М.: Наука, 1977.
12. *Дробышевич В. И., Дымников В. П., Ривин Г. С.* Задачи по вычислительной математике. — М.: Наука, 1980.
13. *Дородницын А. А.* Лекции по численным методам решения уравнений вязкой жидкости. — М.: ВЦ АН СССР, 1969.
14. *Дьяченко В. Ф.* Основные понятия вычислительной математики. — М.: Наука, 1977.
15. *Евтушенко Ю. Г.* Методы решения экстремальных задач и их применение в системах оптимизации. — М.: Наука, 1982.
16. *Зенкевич О.* Метод конечных элементов в технике. — М.: Мир, 1975.
17. *Калиткин Н. Н.* Численные методы. — М.: Наука, 1978.
18. *Кестенбойм Х. С., Росляков Г. С., Чудов Л. А.* Точечный взрыв: Методы расчета. Таблицы. — М.: Наука, 1974.
19. *Карманов В. Г.* Математическое программирование. — М.: Наука, 1986.
20. *Ковеня В. М. Яненко Н. Н.* Методы расщепления в задачах газовой динамики. — Новосибирск: Наука, 1981.

21. *Крылов В. И., Бобков В. В., Монастырный П. И.* Вычислительные методы. Т. 1, 2. — М.: Наука, 1976, 1977.

22. *Лесин В. В., Лисовец Ю. П.* Основы методов оптимизации. — М.: Изд-во МАИ, 1998.

23. *Лифанов И. К.* Метод сингулярных интегральных уравнений и численный эксперимент. — М.: Янус, 1995.

24. *Ляшко И. И., Макаров В. Л., Скоробогатько А. А.* Методы вычислений. — Киев: Высшая школа, 1977.

25. *Магомедов К. М., Холодов А. С.* Сеточно-характеристические численные методы. — М.: Наука, 1988.

26. *Мак-Кракен Д., Дорн У.* Численные методы и программирование на фортране. — М.: Мир, 1977.

27. *Марчук Г. И.* Математические модели в иммунологии. — М.: Наука, 1985.

28. *Марчук Г. И.* Математическое моделирование в проблеме окружающей среды. — М.: Наука, 1982.

29. *Марчук Г. И.* Методы вычислительной математики. — М.: Наука, 1989.

30. *Марчук Г. И.* Численные методы в прогнозе погоды. — Л.: Гидрометеоиздат, 1967.

31. *Марчук Г. И., Лебедев В. И.* Численные методы в теории переноса нейтронов. — М.: Атомиздат, 1971.

32. *Михлин С. Г.* Численная реализация вариационных методов. — М.: Наука, 1968.

33. *На Ц.* Вычислительные методы решения прикладных граничных задач. — М.: Мир, 1982.

34. *Никольский С. М.* Квадратурные формулы. — М.: Наука, 1979.

35. *Одэн Дж.* Конечные элементы в нелинейной механике сплошных сред. — М.: Мир, 1976.

36. *Ортега Дж., Рейнболдт В.* Итерационные методы решения нелинейных систем со многими неизвестными. — М.: Мир, 1975.

37. *Пирумов У. Г.* Численные методы. — М.: Изд-во МАИ, 1998.

38. *Пирумов У. Г., Росляков В. С.* Численные методы газовой динамики. — М.: Высшая школа, 1987.

39. *Победря Б. Е.* Численные методы в теории упругости и пластичности. — М.: Изд-во МГУ, 1981.

40. *Пустыльник Е. И.* Статистические методы анализа и обработки наблюдений. — М.: Наука, 1968.

41. *Пшеничный Б. Н., Данилин Ю. М.* Численные методы в экстремальных задачах. — М.: Наука, 1975.

42. *Рихтмайер Р., Мортон К.* Разностные методы решения краевых задач. — М.: Мир, 1972.

43. *Рождественский Б. Л., Яненко Н. Н.* Системы квазилинейных уравнений и их приложения к газовой динамике. — М.: Наука, 1978.

44. *Роуч П.* Вычислительная гидродинамика. — М.: Мир, 1980.

45. *Рябенький В. С.* Введение в вычислительную математику. — М.: Физматлит, 1994.

46. *Рябенький В. С., Филиппов А. Ф.* Об устойчивости разностных уравнений. — М.: Гостехиздат, 1956.

47. *Самарский А. А.* Введение в численные методы. — М.: Наука, 1987.

48. *Самарский А. А.* Теория разностных схем. — М.: Наука, 1983.

49. *Самарский А. А., Гулин А. В.* Устойчивость разностных схем. — М.: Наука, 1973.

50. *Самарский А. А., Гулин А. В.* Численные методы. — М.: Наука, 1989 .

51. *Самарский А. А., Николаев Е. С.* Методы решения сеточных уравнений. — М.: Наука, 1978.

52. *Самарский А. А., Попов Ю. П.* Разностные методы решения задач газовой динамики. — М.: Наука, 1992.

53. Сборник задач по методам вычислений / Под ред. П. И. Монастырного. — М.: Физматлит, 1994.

54. *Сегерлинд Л.* Применение метода конечных элементов. — М.: Мир, 1979.

55. *Соболь И. М.* Численные методы Монте-Карло. — М.: Наука, 1973.

56. *Стечкин С. Б., Субботин Ю. Н.* Сплайны в вычислительной математике. — М.: Наука, 1976.

57. *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач. — М.: Наука, 1986.

58. *Турчак Л. И.* Основы численных методов. — М.: Наука, 1987.

59. *Уилкинсон Дж.* Алгебраическая проблема собственных значений. — М.: Наука, 1970.

60. *Фаддеев Д. К., Фаддеева В. Н.* Вычислительные методы линейной алгебры. — М.: Физматгиз, 1963.

61. *Форсайт Дж., Малькольм М., Моулер К.* Машинные методы математических вычислений. — М.: Мир, 1980.

62. *Хемминг Р. В.* Численные методы. Для научных работников и инженеров. — М.: Наука, 1968.

63. *Худсон Д.* Статистика для физиков. — М.: Мир, 1970.

64. *Чушкин П. И.* Метод характеристик для пространственных сверхзвуковых течений. — М.: ВЦ АН СССР, 1968.

65. *Шуп Т.* Решение инженерных задач на ЭВМ. — М.: Мир, 1982.

66. *Яненко Н. Н.* Метод дробных шагов решения многомерных задач математической физики. — Новосибирск: Наука, 1967.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Учебное издание

ТУРЧАК Леонид Иванович,
ПЛОТНИКОВ Павел Владимирович

ОСНОВЫ ЧИСЛЕННЫХ МЕТОДОВ

Редактор *Е.Ю. Ходан*
Оригинал-макет *В.И. Шутова*

ЛР № 071930 от 06.07.01. Подписано в печать 25.08.01.
Формат 60×90/16. Бумага офсетная. Печать офсетная.
Усл. печ. л. 19. Уч.-изд. л. 20,9. Тираж 3000 экз.
Заказ №

Издательская фирма
«Физико-математическая литература»
117864 Москва, ул. Профсоюзная, 90

Отпечатано с готовых диапозитивов в ПФ «Полиграфист»
160001, г. Вологда, ул. Челюскинцев, 3.
Тел.: (8172) 72-55-31, 72-61-75, факс (8172) 72-60-72.
E-mail: pfpv@vologda.ru http://www.vologda/∼pfpv

ISBN 5-9221-0153-6

L.I. TURCHAK, P.V. PLOTNIKOV

FUNDAMENTALS OF NUMERICAL METHODS

Russian Academy of Sciences
Physical and Mathematical Literature Publishing Company

Moscow, 2002, 304 pages

ABOUT THE AUTHORS

Leonid I.Turchak:

Was born in 1944 (Ternopol Region, Ukraine).

Studies and Qualifications: Gratuade of Lomonosov Moscow State University — 1966, Postgraduate — 1969; Candidate of Sciences (Physics and Mathematics) — 1970; Docent (Associate Professor) — 1973; Doctor of Sciences (Physics and Mathematics) — 1988; Professor — 1992.

Memberships: American Institute of Aeronautics and Astronautics Senior Member — 1988, Associate Fellow — 1990; Russian Academy of Natural Sciences Associate Fellow — 1992, Fellow — 1994. Current positions: Russian Academy of Sciences Computing Centre, Head of the Computational Physics Department and the Computational Fluid Dynamics (CFD) Sector; Director of the CFD Association.

Professional interests: Development of Numerical Methods for Computer Simulation in Engineering and CFD; Applied Problems in Aeronautics and Astronautics; Lecturer on Numerical Methods and Applications, for students and engineers.

Chief and the Leading Researcher for 2 Grants from the Russian Foundation on Basic Research and 7 Cotracts from the Ministry of Defence. Writer of more than 80 scientific publications including 5 books. The main of them are: Fundamental of Numerical Methods, 1987, 322 p.; Numerical Simulation of Strong Vertical Vortices in the Atmosphere, 2000, 144 p. (in collaboration with S.A. Andrianov, I.I. Vasilchenko, V.N. Zabavin, A.T. Onufrijev, M.D. Shcherbin); Numerical Approaches to Aircraft Dynamics under Aerodynamic Interference, 2001, 207 p. (N.A. Baranov, A.S. Belotserkovsky, M.I. Kanevsky).

E-mail: turchak@ccas.ru

Pavel V. Plotnikov:

Was born in 1972 (Tambov City, Russia).

Education: Moscow Institute of Physics and Technology — 1995, Postgraduate — 1998, Candidate of Sciences (Physics and Mathematics, Computing Center of the Russian Academy of Sciences) — 1998.

Current position: Senior Lecturer of the Tambov State Technical University, Applied Mathematics and Mechanics Chair.

Professional interests: Mathematical Modeling in Computational Fluid Dynamics and Meteor Physics; Problems on Interactions of Space Bodies with the Atmosphere; Lecturer on Mathematics, Numerical Methods and Programming.

Grants: participation in projects of the Russian Foundation on Basic Research — 1994–2001.

Main publications: *Plotnikov P.V., Shurshalov L.V.* Mathematical Modeling of the Process of Extremely Intensive Interaction of an Interplanetary Dust Cloud with the Earth's Atmosphere // Solar Syst. Res. 1997. V. 31, № 1; *Plotnikov P.V.* On the Calculation of Interaction Two-Dimensional between a Cloud of Space Dust Particles and the Atmosphere // Comp. Math. Math. Phys. 1998. V. 38, № 11; *Plotnikov P.V., Shurshalov L.V.* Modeling of a catastrophic collision of space dust with a planet atmosphere // Computational Fluid Dynamics Journal. 2001. V. 10, № 3.

E-mail: pvp@fdo.tgtu.tambov.ru

ABSTRACT

This book is addressed to students and specialists as a first course in numerical methods. It is an elementary introduction to the subject, but also contains sufficient material to be useful to readers for simple numerical method applications in solving some problems. The authors have not presupposed a high level of mathematical education for this book. Some indispensable mathematical preliminaries are given before consideration of the numerical methods. The uninterpreted connection between the classic course of mathematics and its computer section is tracing. To give a better understanding of the nature of the numerical methods discussed in the text, a large number of exercises and algorithm schemes are provided. The exposition is weighted from the point of view of using computers, and the reader will be able to program the algorithms without difficulty.

The book contains nine chapters. The first of them is devoted to the accuracy of computational experiments. Sources of errors in calculations using computers are considered. The elementary theory of errors is given. Function approximations and their applications are given in Ch. 2. Some types of interpolation including the spline interpolation are considered. Some informative material related to practical use is provided. In Ch. 3 numerical methods for differentiation and integration are given. Numerical differentiation is important for working out difference schemes, and some required relations are given here. An adaptive algorithm strategy for numerical integration is considered. Problems in linear algebra and their numerical solutions are contained in Ch. 4. The problem of calculating eigenvalues is also described. The most widely used direct and iterative methods for the solution of linear systems are expounded. Non-linear equations and their systems are considered in Ch. 5. Methods for both algebraic and transcendent equations are given. Optimization methods are given in Ch. 6. The problem of linear programming is also considered here. Methods for the numerical solution of ordinary differential equations are considered in Ch. 7. There are two types of the problem here: the initial problem and the boundary value problem. The main

numerical methods are discussed here. Certain partial differential equations and some numerical methods for their calculation are considered in Ch. 8. Some concepts of difference schemes are given here. The last, Ch. 9 is devoted to the integral equations. Here singular integrals and equations are also considered.

This book has been approved and recommended for publishing by some well-known scientists who are specialists in numerical methods, their applications and teaching. The Ministry of Higher Education has confirmed this book as a suitable textbook for students.

CONTENTS