Л.И. ТУРЧАК

ОСНОВЫ ЧИСЛЕННЫХ МЕТОДОВ

Л. И. ТУРЧАК

ОСНОВЫ ЧИСЛЕННЫХ МЕТОДОВ

Под редакцией В. В. ЩЕННИКОВА

Допущено Министерством высшего и среднего специального образования СССР в качестве учебного пособия для студентов высших учебных заведений

МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
1987

ББК 22.19
Т89
УДК 519.6(075.8)

Турчак Л. И. **Основы численных методов:** Учеб. пособие.— М.: Наука. Гл. ред. физ.-мат. лит., 1987.— 320 с.

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 блок-схем для облегчения понимания логической структуры рассматриваемых методов и их использования в расчетах на ЭВМ.

Для студентов вузов.

Табл. 20. Ил. 72. Библиогр. 56 назв.

Рецензенты:

кафедра вычислительной математики Московского физико-технического института,

кафедра общей и прикладной математики Завода-втуза при Московском автомобильном заводе им. И. А. Лихачева,

доктор физико-математических наук *В. П. Шидловский*

Т $\frac{1702070000-022}{053(02)-87}$ 65-86

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Внедрение ЭВМ 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 ЭВМ и простейших численных методов, не говоря уже о том, что при выполнении курсовых и дипломных работ применение вычислительных машин становится нормой в подавляющем большинстве вузов.

Вычислительная техника используется сейчас не только в инженерных и экономических науках, но и в таких традиционно нематематических специальностях, как медицина, лингвистика, психология и др. В связи с этим можно констатировать, что применение ЭВМ приобрело массовый характер. Возникла многочисленная категория специалистов — пользователей ЭВМ, для которых необходима литература по дисциплинам, непосредственно связанным с применением вычислительной техники.

Основной такой дисциплиной 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 главной задачей 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 блок-схемы, а также примеры решения задач, способствующие лучшему пониманию материала. Книга написана с учетом особенностей применения численных методов при решении задач с использованием ЭВМ.

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 автор использовал в работе над книгой. 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 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, которые студенты могут использовать при выполнении расчетных заданий.

Книга содержит девять глав. В гл. 1 излагаются основные понятия, связанные с погрешностями вычислений. Рассматриваются источники погрешностей при расчетах на ЭВМ.

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 гл. 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 гл. 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 гл. 7.

В гл. 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.).

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

ВВЕДЕНИЕ

1. Этапы решения задачи на ЭВМ. Наиболее эффективное применение вычислительная техника нашла при проведении трудоемких расчетов в научных исследованиях. Действительно, современные ЭВМ за 1 мс выполняют такой объем вычислений, на который человеку понадобится целый день.

При решении задачи на ЭВМ 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 перечисленные здесь 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 ЭВМ может выполнять лишь простейшие операции, она «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 и построение блок-схемы. Процесс решения задачи (вычислительный процесс) записывается в виде последовательности элементарных арифметических и логических операций, приводящей к конечному результату и называемой *алго-*

ритмом решения задачи. Алгоритм можно изобразить в виде блок-схемы.

Программирование. Алгоритм решения задачи записывается на понятном машине языке в виде точно определенной последовательности операций — *программы* для ЭВМ. Составление программы (программирование) обычно производится с помощью некоторого промежуточного (алгоритмического) языка, а ее трансляция (перевод на язык ЭВМ) осуществляется самой вычислительной системой.

Отладка программы. 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.

Проведение расчетов. На этом этапе готовятся исходные данные для расчетов и проводится счет по отлаженной программе. При этом для уменьшения ручного труда по обработке результатов можно широко использовать удобные формы выдачи результатов, например распечатку таблиц, построение графиков.

Анализ результатов. Результаты расчетов тщательно анализируются, оформляется научно-техническая документация.

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

Не умаляя значения физического эксперимента, нужно все-таки отметить неуклонно возрастающую долю вычислений на ЭВМ 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).

Пример. 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

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

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, о движении планера и т. п.). Заметим также, что модель (0.1) легко получается из (0.2) 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 и т. д. 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 ЭВМ.

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: графические, аналитические и численные.

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.

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 слишком редкие случаи.

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. Многие численные методы разработаны давно, однако при вычислениях вручную они могли использоваться лишь для решения не слишком трудоемких задач.

С появлением ЭВМ начался период бурного развития численных методов и их внедрения в практику. Только вычислительной машине под силу выполнить за сравнительно короткое время объем вычислений в миллионы, миллиарды и более операций, необходимых для решения многих современных задач. При счете вручную человеку не хватило бы и жизни для решения одной такой задачи.

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 I

ТОЧНОСТЬ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА

§ 1. Приближенные числа

1. Числа с плавающей точкой. ЭВМ обрабатывают числа, которые записаны в формах с фиксированной точкой и плавающей точкой *).

Десятичные числа с *фиксированной точкой* — это привычная нам форма записи чисел: 5, —10, 175.12, 0.0093 и т. п.; здесь вместо десятичной запятой ставится точка.

Как известно, множество целых чисел бесконечно. Однако ЭВМ из-за ограниченности ее разрядной сетки может оперировать лишь с некоторым конечным подмножеством этого множества. Так, во многих моделях ЭВМ диапазон представляемых целых чисел даже в режиме с удвоенной точностью находится примерно в интервале от $-2 \cdot 10^9$ до $2 \cdot 10^9$.

При решении научно-технических задач в основном используются действительные (вещественные) числа. Для их представления почти во всех машинах используется форма 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 $m = 0,d_1d_2 \ldots d_k$, 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.

Все сказанное выше распространяется и на числа, записанные в других системах счисления. Число N в системе счисления с основанием α можно представить в виде

*) Термины «фиксированная точка» и «плавающая точка» широко используются в операционных системах ЭВМ. Понятия «фиксированная запятая» и «плавающая запятая» утратили силу (применительно к ЭВМ).

$N = \pm 0.a_1a_2 \ldots a_k \cdot \alpha^n$. Из этой записи следует, что подмножество действительных чисел, с которым оперирует конкретная ЭВМ, 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 $2(\alpha - 1)(n_2 - n_1 + 1)\alpha^{k-1} + 1$ чисел.

Границы порядка n_1, n_2 определяют ограниченность действительных чисел по величине, а размерность k — дискретность распределения их на отрезке числовой оси. Например, в случае десятичных чисел при четырехразрядном представлении все значения, находящиеся в интервале между числами 0.2851 и 0.2852, представляются числом 0.2851 (при отбрасывании остальных разрядов без округления). Разность между двумя соседними значениями равна единице последнего разряда. Числа, меньшие этой разности, воспринимаются как машинный нуль.

Таким образом, ЭВМ оперируют 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. На-

пример, значение $a = 0.734$ могло быть получено округлением чисел 0.73441, 0.73353 и др. При этом $|\Delta x| \leqslant \leqslant 0.0005$, и полагаем $\Delta a = 0.0005$.

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

При вычислениях на ЭВМ округления, как правило, не производятся, а цифры, выходящие за разрядную сетку машины, отбрасываются. В этом случае максимально возможная погрешность результата выполнения операции в два раза больше по сравнению со случаем округления.

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 $0.110 \cdot 10^2 = 11$ 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. Относительная погрешность суммы заключена между наибольшим и наименьшим значениями относительных погрешностей слагаемых; на практике принимается наибольшее значение.

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:

$$\Delta(a \pm b) = \Delta a + \Delta b, \quad \delta(a \cdot b) = \delta a + \delta b, \qquad \delta(a/b) = \delta a + \delta b, \quad \delta(a^k) = k\delta a. \tag{1.1}$$

Пример 1. Найти относительную погрешность функции

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

Используя формулы (1.1), получаем

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

Полученная оценка относительной погрешности содержит в знаменателе выражение $|1-x|$. Ясно, что при $x \approx 1$ можем 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. Не-

трудно 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 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

$$\Delta u = |f'_x(a, b, c)|\,\Delta a + |f'_y(a, b, c)|\,\Delta b + |f'_z(a, b, c)|\,\Delta c, \tag{1.2}$$

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

$$\delta u = \frac{\Delta u}{|f(a, b, c)|}. \tag{1.3}$$

Полученные соотношения можно использовать для вывода оценки погрешности произвольной функции (таким способом легко получить выражения (1.1)). Например, при $c = a - b$ по формуле (1.2) 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 ЭВМ 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. Это так называемые *неустранимые погрешности*, 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, усечением рядов при вычислениях значений функций, интерполированием табличных данных и т. п. Как правило, *погрешность численного метода* регулируема, т. е. она может быть уменьшена до любого разумного значения путем изменения некоторого параметра (напри-

мер, шага интегрирования, числа членов усеченного ряда и т. п.). Погрешность метода обычно стараются довести до величины, в несколько раз меньшей погрешности исходных данных. 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 ЭВМ неизбежны *погрешности округлений*, связанные с ограниченностью разрядной сетки машины. Обычно после выполнения операции производится не округление результата, а простое отбрасывание лишних разрядов с целью экономии машинного времени. Правда, в современных машинах предусмотрена свобода выбора программистом способа округления; соответствующими средствами располагают и некоторые алгоритмические языки (например, кобол, ПЛ-1).

Максимальная относительная погрешность при округлении есть $\delta_{\max} = 0.5\alpha^{1-k}$, где α — основание системы счисления, k — количество разрядов мантиссы числа. При простом отбрасывании лишних разрядов эта погрешность увеличивается вдвое.

В современных машинах с памятью, измеряемой в байтах, принята шестнадцатеричная система счисления, и любое число с плавающей точкой содержит шесть значащих цифр. Следовательно, $\alpha = 16$, $k = 6$, максимальная погрешность округления $\delta_{\max} = 0.5 \cdot 16^{-5} \approx 0.5 \cdot 10^{-6}$.

Несмотря на то что при решении больших задач выполняются миллиарды 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 такие случаи рассмотрим.

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 $0.1 = 0.000\,1100\,1100\ldots$ 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 машине округление происходит после каждого сложения. Предполагая условно сетку четырехразрядной, проследим вычисление на машине 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. 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 ЭВМ нужно использовать подобного рода «маленькие хитрости» для улучшения алгоритма и снижения погрешностей результатов.

Например, при вычислении на ЭВМ 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 $(a+x)^2 = a^2 + 2ax + x^2$.

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

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

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

$$\sin 0.5236 \approx 0.5236 - 0.2393 \cdot 10^{-1} + 0.3281 \cdot 10^{-3} = 0.500.$$

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 $\sin 6.807 \approx 0.5493$. Относительная погрешность составляет здесь около 10% (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 $x = 25.66$ $(1470° = 4 \cdot 360° + 30°)$ 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: $\sin 25.66 \approx 24$.

В программах, использующих степенные ряды для вычисления значений функций, могут быть приняты различные меры по предотвращению подобной потери точности. Так, 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 ($e^x = e^{n+a} = e^n e^a$) и использовать разложение в ряд только для e^a, а e^n вычислять умножением. Таким образом, при организации вычислений можно своевременно распознать подобные «подводные камни», когда возможна потеря точности, и попытаться затем исправить положение.

3. О решении квадратного уравнения. Мы убедились в том, что при численном решении задач на ЭВМ 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

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_2 = \frac{-b + \sqrt{D}}{2a}, \quad D = b^2 - 4ac. \quad (1.4)$$

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. (1.4)); $D = 0$ — 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 (1.4) 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:

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

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

Более универсальным способом является использование значения sign b («знак величины b»):

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

Тогда один из корней может быть вычислен по формуле

$$x_1 = -(b + \operatorname{sign} b \cdot \sqrt{D})/(2a). \qquad (1.5)$$

Выражение для вычисления значения второго корня можно получить следующим путем. Представим квадратное уравнение в виде

$$ax^2 + bx + c = a(x - x_1)(x - x_2) =$$
$$= ax^2 - ax(x_1 + x_2) + ax_1x_2.$$

Приравнивая свободные члены, получаем

$$x_2 = c/(ax_1). \qquad (1.6)$$

На рис. 1 представлен один из вариантов блок-схемы алгоритма решения квадратного уравнения с учетом рассмотренных здесь особенностей. При $D > 0$ значения корней вычисляются по формулам (1.5), (1.6). Заметим, что в приведенном на блок-схеме 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 ЭВМ невозможна.

Примеры. 1. $a = 10^{-40}$, $b = -3 \cdot 10^{-40}$, $c = 2 \cdot 10^{-40}$.

При вычислении произведений b^2 и $4ac$ получается машинный нуль, т. е. $D = 0$; решение пойдет по ветви равных корней: $x_1 = x_2 = 1.5$. Точные значения корней, как нетрудно видеть, равны $x_1 = 1$, $x_2 = 2$.

2. $a = 10^{40}$, $b = -3 \cdot 10^{40}$, $c = 2 \cdot 10^{40}$;

Этот вариант аналогичен предыдущему случаю с той лишь разницей, что вместо получения машинного нуля произойдет переполнение и прерывание счета.

3. $a = 10^{-40}$, $b = 10^{10}$, $c = -10^{40}$.

Это трудный для реализации на ЭВМ 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.

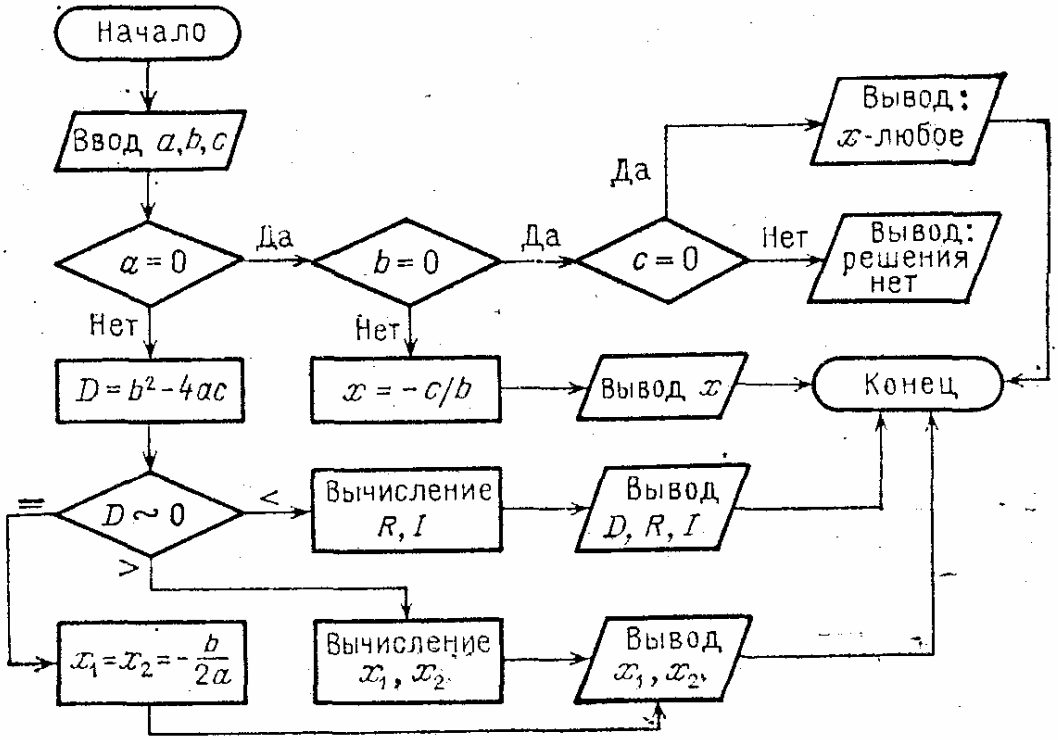


Рис. 1. Блок-схема решения квадратного уравнения

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 впра-

ве 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 результате расчетов.

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

Примером такой задачи является отыскание действительных корней многочленов вида

$$(x-a)^n = \varepsilon, \quad 0 < \varepsilon \ll 1.$$

Изменение правой части на величину порядка ε приводит к погрешности корней порядка $\varepsilon^{1/n}$.

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

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

Очевидно, что корнями этого многочлена являются $x_1 = 1$, $x_2 = 2$, ..., $x_{20} = 20$.

Предположим, что один из коэффициентов многочлена вычислен с некоторой малой погрешностью. Например, коэффициент -210 при x^{19} увеличим на 2^{-23} (около 10^{-7}). В результате вычислений даже с точностью до 11 значащих цифр получим существенно другие значения корней. Приведем для наглядности эти значения, округленные до трех знаков:

$$x_1 = 1.00, \qquad x_9 = 8.92,$$
$$x_2 = 2.00, \qquad x_{10,11} = 10.1 \pm 0.644i,$$

$$x_3 = 3.00, \quad x_{12,\,13} = 11.8 \pm 1.65i,$$
$$x_4 = 4.00, \quad x_{14,\,15} = 14.0 \pm 2.52i,$$
$$x_5 = 5.00, \quad x_{16,\,17} = 16.7 \pm 2.81i,$$
$$x_6 = 6.00, \quad x_{18,\,19} = 19.5 \pm 1.94i,$$
$$x_7 = 7.00, \quad x_{20} = 20.8.$$
$$x_8 = 8.01,$$

Таким образом, изменение коэффициента -210 при x^{19} на $-210 + 10^{-7}$ привело к тому, что половина корней стали комплексными. Причина такого явления — неустойчивость самой задачи; вычисления выполнялись очень точно (11 разрядов), а погрешности округлений не могли привести к таким последствиям.

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

Рассмотренные выше примеры неустойчивых задач являются некорректно поставленными. Применять для решения таких задач численные методы, как правило, нецелесообразно, поскольку возникающие в расчетах погрешности округлений 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

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

Интегрируя по частям, находим

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

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

. . . . . .

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

Пользуясь полученным рекуррентным соотношением, вычисляем

$$I_1 = 0.367879, \quad I_6 = 0.127120,$$
$$I_2 = 0.263242, \quad I_7 = 0.110160,$$
$$I_3 = 0.207274, \quad I_8 = 0.118720,$$
$$I_4 = 0.170904, \quad I_9 = -0.0684800.$$
$$I_5 = 0.145480,$$

Значение интеграла I_9 не может быть отрицательным, поскольку подынтегральная функция x^9e^{x-1} на всем отрезке интегрирования [0, 1] неотрицательна. Исследуем источник погрешности. Видим, что округление в I_1 дает погрешность, равную примерно лишь $4.4 \cdot 10^{-7}$. Однако на каждом этапе эта погрешность умножается на число, модуль которого больше единицы (—2, —3, ..., —9), что в итоге дает 9!. 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 нецелесообразно; к ним будем обращаться при изучении численных методов.

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

Упражнения

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

2. Записать в форме с фиксированной точкой числа $0.312\cdot10^3$, $-0.70\cdot10^1$, $0.465\cdot10^{-2}$.

3. Указать максимально возможные абсолютные и относительные погрешности приближенных чисел 27, —14.0, 0.00173, $0.745\cdot10^{-4}$, $-0.245\cdot10^4$, $-0.8960\cdot10^2$.

4. Оценить погрешности величин x, y, заданных соотношениями

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

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

5. Найти относительные погрешности при вычислении определителей

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

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

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 $y = f(x)$. В некоторых случаях даже при известной зависимости $y = f(x)$ она настолько громоздка (например, содержит трудно вычисляемые выражения, сложные интегралы и т. п.), что ее использование в практических расчетах затруднительно.

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 $y = f(x)$ неизвестна.

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*.

Для практики весьма важен случай аппроксимации функции многочленом

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

В дальнейшем будем рассматривать лишь такого рода аппроксимацию. 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.

Если приближение строится на заданном дискретном множестве точек $\{x_i\}$, то аппроксимация называется *точечной*. К ней относятся интерполирование, среднеквадратичное приближение и др. При построении приближения на непрерывном множестве точек (например, на отрезке $[a, b]$) аппроксимация называется *непрерывной* (или *интегральной*).

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

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

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*, а многочлен $\varphi(x)$ — *интерполяционным многочленом*.

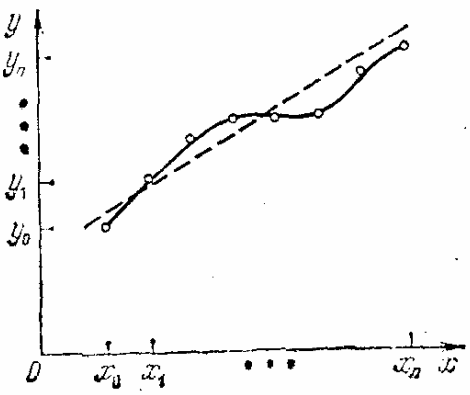


Рис. 2. Интерполяция и аппроксимация

Таким образом, близость интерполяционного многочлена к заданной функции состоит в том, что их значения совпадают на заданной системе точек (рис. 2, сплошная линия).

Максимальная степень интерполяционного многочлена $m = n$; в этом случае говорят о *глобальной интерполяции*, поскольку один многочлен

$$\varphi(x) = a_0 + a_1 x + \ldots + a_n x^n \qquad (2.3)$$

используется для интерполяции функции $f(x)$ на всем

рассматриваемом интервале изменения аргумента x. Коэффициенты a_j многочлена (2.3) находятся из системы уравнений (2.2). Можно показать, что при $x_i \neq x_k$ $(i \neq k)$ эта система имеет единственное решение.

Интерполяционные многочлены могут строиться отдельно для разных частей рассматриваемого интервала изменения x. В этом случае имеем *кусочную* (или *локальную*) *интерполяцию*.

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

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

Например, при большом количестве узлов интерполяции получается высокая степень многочлена (2.3) в случае глобальной интерполяции, т. е. когда нужно иметь один интерполяционный многочлен для всего интервала изменения аргумента. Кроме того, табличные данные могли быть получены путем измерений и содержать ошибки. Построение аппроксимирующего многочлена с условием обязательного прохождения его графика через эти экспериментальные точки означало бы тщательное повторение допущенных при измерениях ошибок. Выход из этого положения может быть найден выбором такого многочлена, график которого проходит близко от данных точек (см. рис. 2, штриховая линия). Понятие «близко» уточняется при рассмотрении разных видов приближения.

Одним из таких видов является *среднеквадратичное приближение* функций с помощью многочлена (2.1). При этом $m \leqslant n$; случай $m = n$ соответствует интерполяции. На практике стараются подобрать аппроксимирующий многочлен как можно меньшей степени (как правило, $m = 1, 2, 3$).

Мерой отклонения многочлена $\varphi(x)$ 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 многочлена и функции в данных точках:

$$S = \sum_{i=0}^{n} [\varphi(x_i) - y_i]^2. \qquad (2.4)$$

Для построения аппроксимирующего многочлена нужно подобрать коэффициенты a_0, a_1 ..., a_m так, чтобы величина S была наименьшей. В этом состоит *метод наименьших квадратов.*

3. 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 многочлена $\varphi(x)$ 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 многочлен $\varphi(x)$ *равномерно аппроксимирует* функцию $f(x)$ с точностью ε на отрезке $[a, b]$.

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

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

По аналогии можно ввести понятие *среднеквадратичного отклонения* $\overline{\Delta} = \sqrt{S/n}$ при среднеквадратичном приближении функций. На рис. 3 показано принципиальное различие двух рассматриваемых приближений.

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

Теорема. *Если функция $f(x)$ непрерывна на отрезке $[a, b]$, то для любого $\varepsilon > 0$ существует многочлен $\varphi(x)$ 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 ЭВМ значений элементарных функций.

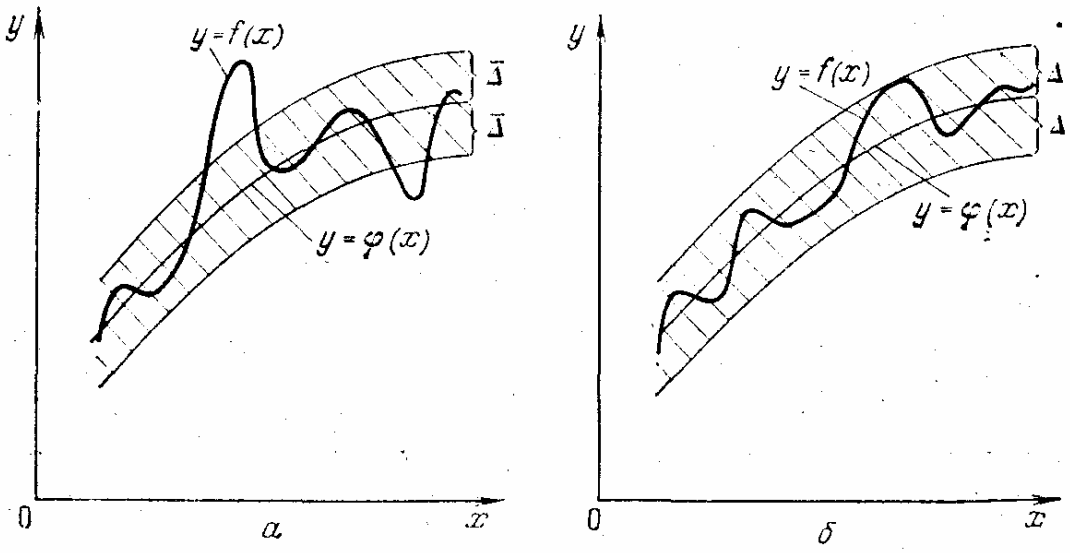


Рис. 3. Приближения: *а* — среднеквадратичное; *б* — равномерное

Существует также понятие *наилучшего приближения функции* $f(x)$ многочленом $\varphi(x)$ фиксированной степени m. В этом случае коэффициенты многочлена $\varphi(x) = a_0 + a_1x + \ldots + a_mx^m$ следует выбрать так, чтобы на заданном отрезке $[a, b]$ величина абсолютного отклонения (2.5) была минимальной. Многочлен $\varphi(x)$ называется *многочленом наилучшего равномерного приближения*. Существование и единственность многочлена наилучшего равномерного приближения вытекает из следующей теоремы.

Теорема. *Для любой функции* $f(x)$, *непрерывной на замкнутом ограниченном множестве* G, *и любого натурального* m *существует многочлен* $\varphi(x)$ *степени не выше* m, *абсолютное отклонение которого от функции* $f(x)$ *минимально, т. е.* $\Delta = \Delta_{\min}$, *причем такой многочлен единственный.*

Множество G обычно представляет собой либо некоторый отрезок $[a, b]$, либо конечную совокупность точек $x_0, x_1, \ldots, x_n$.

§ 2. Использование рядов

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

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \qquad (2.6)$$

При известном значении аргумента x значение функции может быть получено с точностью до погрешностей округления. Количество используемых членов ряда (2.6) 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 (2.6) с учетом соотношения $\cos x = \sin(\pi/2 + x)$. Гиперболические синус и косинус можно вычислить с помощью разложения в ряд экспоненты e^x, поскольку

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

Правда, здесь есть опасность вычитания близких чисел при вычислении $\operatorname{sh} x$ для $x \approx 0$, что приведет к потере точности. В таких случаях лучше воспользоваться рядом

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

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

В качестве примера построим алгоритм вычисления синуса с помощью ряда (2.6). 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.

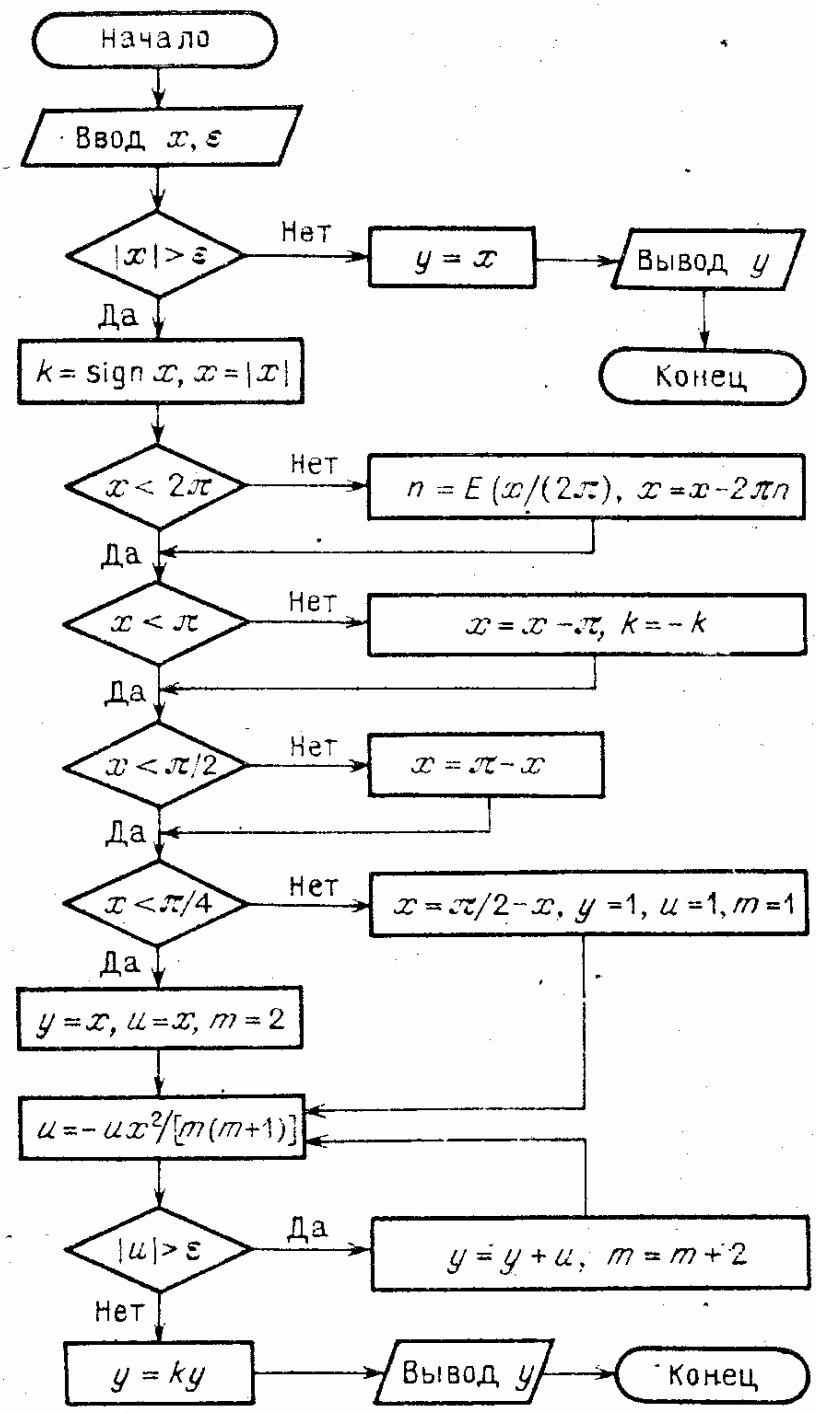


Рис. 4. Блок-схема вычисления синуса

В этом случае учитываются все члены, большие машинного нуля, а точность результата определяется погрешностями округлений.

Возможный вариант алгоритма вычисления синуса с помощью ряда (2.6) изображен на рис. 4 в виде блок-схемы. Дадим некоторые пояснения к ней.

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

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

При анализе точности вычислений отмечалось, что при суммировании ряда погрешность значительно меньше, если $|x| < 1$. Поэтому в блок-схеме 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: $\sin(\pi \pm x) = \mp \sin x$, $\sin(\pi/2 - x) = \cos x$. Например, при $x = 7.6\,\pi$ $(k = 1)$ получим следующий алгоритм:

$$n = E(7.6\pi/(2\pi)) = E(3.8) = 3,$$

$$x - 2\pi n = 7.6\pi - 6\pi = 1.6\pi > \pi, \quad k = -1,$$

$$x - \pi = 0.6\pi > \pi/2, \quad \pi - x = 0.4\pi > \pi/4, \quad \pi/2 - x = 0.1\pi.$$

Текущее значение члена ряда в блок-схеме обозначено через u, значение функции — через y. Здесь используется выражение каждого члена ряда через предшествующий. Например,

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

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

$$\sin\left(\frac{\pi}{2} - x\right) = \cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$$

Для этого полагаем $y = 1$, $u = 1$, $m = 1$. В остальном алгоритм не меняется.

В рассмотренном алгоритме аргумент предполагается заданным в радианах. Если он задан в градусах, то в блок-схеме следует предусмотреть оператор перевода в радианы, т. е. умножения на величину $\pi/180$.

Погрешность функции $y = \sin x$, полученной с помощью ряда (2.6) с использованием приведенного алго-

ритма (см. рис. 4), состоит из двух частей — погрешности округления и *погрешности ограничения*, возникающей из-за учета лишь ограниченного числа членов ряда.

Погрешности ограничений зависят от значения аргумента. При $|x| < \pi/4$ они весьма малы и сравнимы с погрешностями округлений, а с увеличением x возрастают. В частности, если ограничиться первыми пятью членами разложения (2.6) и провести вычисления с точностью до восьми разрядов, то максимальная погрешность составит около $4 \cdot 10^{-6}$ (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:

$$T_n(x) = \frac{1}{2}\left[(x + \sqrt{x^2-1})^n + (x - \sqrt{x^2-1})^n\right],$$

$$-1 \leqslant x \leqslant 1, \quad n = 0, 1, \ldots \qquad (2.7)$$

Легко показать, что (2.7) действительно является многочленом: при возведении в степень и последующих преобразованиях члены, содержащие корни, уничтожаются. Приведем многочлены Чебышева, полученные по формуле (2.7) при $n = 0, 1, 2, 3$ (рис. 5):

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

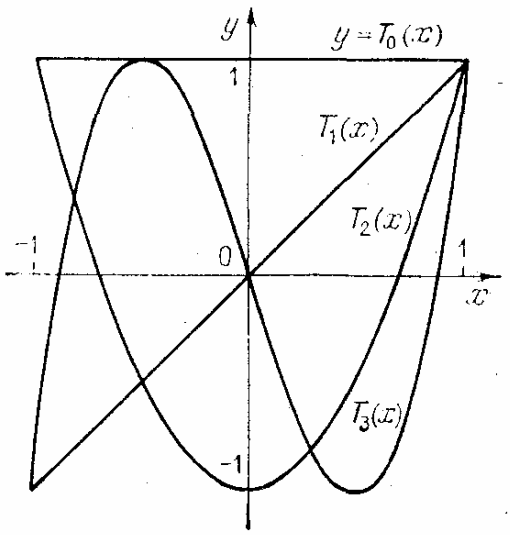


Рис. 5. Многочлены Чебышева

Для вычисления многочленов Чебышева можно воспользоваться рекуррентным соотношением

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \ldots \qquad (2.8)$$

В ряде случаев важно знать коэффициент a_n при стар-

шем 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

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

Перейдем к пределу при $x \to \infty$ и воспользуемся формулой (2.7). Получим

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

Многочлены Чебышева можно также представить в тригонометрической форме:

$$T_n(x) = \cos(n \arccos x), \quad n = 0, 1, \ldots \qquad (2.9)$$

С помощью этих выражений могут быть получены формулы (2.7), (2.8).

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

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

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 (2.6) 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 (2.6) для аппроксимации синуса на отрезке $[-1, 1]$ примет вид

$$\sin\frac{\pi x}{2} = \frac{\pi x}{2} - \frac{1}{3!}\left(\frac{\pi x}{2}\right)^3 + \frac{1}{5!}\left(\frac{\pi x}{2}\right)^5 - \ldots \qquad (2.10)$$

При использовании этого ряда погрешность вычисления функции в окрестности концов отрезка $x = \pm 1$ существенно возрастает и становится значительно больше, чем в окрестности точки $x = 0$. Если вместо (2.10) использовать ряд

$$\sin(\pi x/2) = c_0 + c_1 T_1(x) + c_2 T_2(x) + \ldots,$$

членами которого являются многочлены Чебышева, то погрешность будет распределена равномерно по всему отрезку (рис. 6). В частности, при использовании многочленов Чебышева до девятой степени включительно погрешность находится в интервале $(-5 \div 5) \cdot 10^{-9}$. Для сравнения напомним, что погрешность ряда Тейлора для этой задачи на концах отрезка составляет $4 \cdot 10^{-6}$.

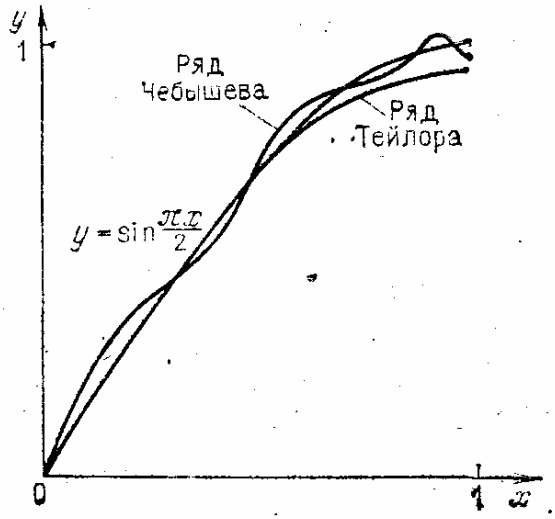


Рис. 6

Нахождение коэффициентов ряда Чебышева довольно сложно и здесь рассматриваться не будет. На практике часто используют многочлены Чебышева для повышения точности аппроксимации функций с помощью ряда Тейлора.

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

$$f(x) \approx a_0 + a_1 x + \ldots + a_n x^n. \qquad (2.11)$$

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

заменой переменной

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

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

$$x^n = -2^{1-n}(b_0 + b_1 x + \ldots + b_{n-1} x^{n-1}) + 2^{1-n} T_n(x). \quad (2.12)$$

Если отбросить последний член, то допущенную при этом погрешность Δ легко оценить: $|\Delta| \leqslant 2^{1-n}$, поскольку $|T_n(x)| \leqslant 1$. Таким образом, из (2.12) получаем, что x^n есть линейная комбинация более низких степеней x. Подставляя эту линейную комбинацию в (2.11), приходим к многочлену степени $n-1$ вместо многочлена степени n. Этот процесс может быть продолжен до тех пор, пока погрешность не превышает допустимого значения.

Используем эту процедуру для повышения точности аппроксимации функции с помощью ряда (2.10). Будем учитывать члены ряда до 11-й степени включительно. Вычисляя коэффициенты при степенях x, получаем

$$\sin(\pi x/2) \approx 1.5707963x - 0.64596410x^3 + 0.079692626x^5 - 0.0046817541x^7 + 0.00016044118x^9 - 0.0000035988432x^{11}. \quad (2.13)$$

Многочлен Чебышева 11-й степени имеет вид

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

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

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

Подставляя в (2.13) вместо x^{11} правую часть этого равенства и вычисляя новые значения коэффициентов, получаем

$$\sin(\pi x/2) \approx 1.5707962x - 0.64596332x^3 + 0.079688296x^5 - 0.0046718573x^7 + 0.00015054436x^9 - 0.00000000351T_{11}. \quad (2.14)$$

Отбрасывая последний член этого разложения, мы допускаем погрешность $|\Delta| \leqslant 3.51 \cdot 10^{-9}$. Из-за приближенного вычисления коэффициентов при степенях x реаль-

ная погрешность больше. Здесь она оценивается величиной $|\Delta| \leqslant 8 \cdot 10^{-9}$. Эта погрешность немного больше, чем для многочлена Чебышева ($5 \cdot 10^{-9}$), и значительно меньше, чем для ряда Тейлора ($4 \cdot 10^{-6}$).

Процесс модификации приближения можно продолжить. Если допустимое значение погрешности больше, чем при использовании выражения (2.14) (без последнего члена с T_{11}), то x^9 можно заменить многочленом седьмой степени, а член с T_9 отбросить; так продолжать до тех пор, пока погрешность остается меньше допустимой.

В заключение приведем некоторые формулы, необходимые при использовании многочленов Чебышева.

1. Многочлены Чебышева:

$$T_n(x) = \frac{1}{2}\left[(x + \sqrt{x^2-1})^n + (x - \sqrt{x^2-1})^n\right] = \cos(n \arccos x),$$

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

$$T_0(x) = 1,$$

$$T_1(x) = x,$$

$$T_2(x) = 2x^2 - 1,$$

$$T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1,$$

$$T_5(x) = 16x^5 - 20x^3 + 5x,$$

$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1,$$

$$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x,$$

$$T_8(x) = 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1,$$

$$T_9(x) = 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x,$$

$$T_{10}(x) = 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1,$$

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

2. Представление степеней x через многочлены $T_n(x)$:

$$x^0 = 1 = T_0,$$

$$x = T_1,$$

$$x^2 = \frac{1}{2}(T_0 + T_2),$$

$$x^3 = \frac{1}{4}(3T_1 + T_3),$$

$$x^4 = \frac{1}{8}(3T_0 + 4T_2 + T_4),$$

$$x^5 = \frac{1}{16}(10T_1 + 5T_3 + T_5),$$

$$x^6 = \frac{1}{32}(10T_0 + 15T_2 + 6T_4 + T_6),$$

$$x^7 = \frac{1}{64}(35T_1 + 21T_3 + 7T_5 + T_7),$$

$$x^8 = \frac{1}{128}(35T_0 + 56T_2 + 28T_4 + 8T_6 + T_8),$$

$$x^9 = \frac{1}{256}(126T_1 + 84T_3 + 36T_5 + 9T_7 + T_9),$$

$$x^{10} = \frac{1}{512}(126T_0 + 210T_2 + 120T_4 + 45T_6 + 10T_8 + T_{10}),$$

$$x^{11} = \frac{1}{1024}(462T_1 + 330T_3 + 165T_5 + 55T_7 + 11T_9 + T_{11}).$$

3. Выражение x^n через более низкие степени:

$$x = T_1,$$

$$x^2 = \frac{1}{2}(1 + T_2),$$

$$x^3 = \frac{1}{4}(3x + T_3),$$

$$x^4 = \frac{1}{8}(8x^2 - 1 + T_4),$$

$$x^5 = \frac{1}{16}(20x^3 - 5x + T_5),$$

$$x^6 = \frac{1}{32}(48x^4 - 18x^2 + 1 + T_6),$$

$$x^7 = \frac{1}{64}(112x^5 - 56x^3 + 7x + T_7),$$

$$x^8 = \frac{1}{128}(256x^6 - 160x^4 + 32x^2 - 1 + T_8),$$

$$x^9 = \frac{1}{256}(576x^7 - 432x^5 + 120x^3 - 9x + T_9),$$

$$x^{10} = \frac{1}{512}(1280x^8 - 1120x^6 + 400x^4 - 50x^2 + 1 + T_{10}),$$

$$x^{11} = \frac{1}{1024}(2816x^9 - 2816x^7 + 1232x^5 - 220x^3 + 11x + T_{11}).$$

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

$$P(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n. \qquad (2.15)$$

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, как это сделано при вычислении синуса (см. рис. 4), удается выразить каждый последующий член через предыдущий и таким образом значительно сократить объем вычислений.

Анализ многочлена (2.15) в общем случае приводит к тому, что для исключения возведения x в степень в каждом члене многочлен целесообразно переписать в виде

$$P(x) = a_0 + x(a_1 + x(a_2 + \ldots + x(a_{n-1} + xa_n) + \ldots)). \qquad (2.16)$$

Прием, с помощью которого многочлен представляется в таком виде, называется *схемой Горнера*. Соответствующий ему алгоритм вычисления значения многочлена изображен на рис. 7. 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.

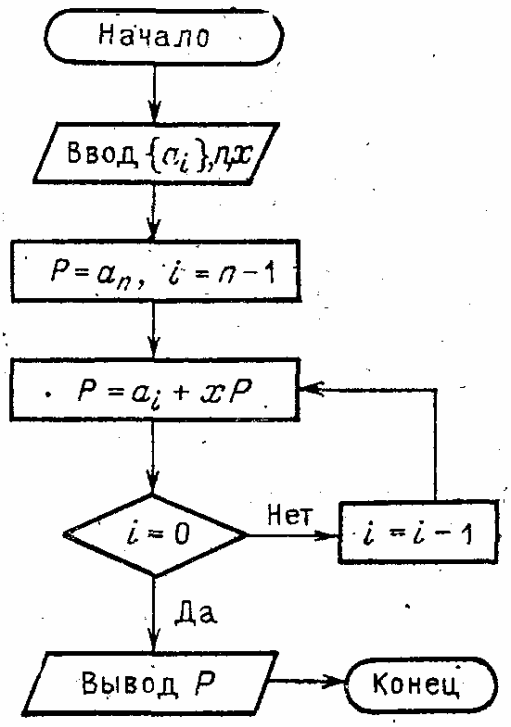


Рис. 7. Блок-схема метода Горнера

4. 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:

$$f(x) = \frac{b_0 + b_1 x + b_2 x^2 + b_3 x^3}{1 + c_1 x + c_2 x^2 + c_3 x^3}. \qquad (2.17)$$

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.

Перепишем выражение (2.17) в виде

$$b_0 + b_1 x + b_2 x^2 + b_3 x^3 = (1 + c_1 x + c_2 x^2 + c_3 x^3) f(x).$$

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

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots \qquad (2.18)$$

и учитывая члены до шестой степени включительно, получаем

$$b_0 + b_1 x + b_2 x^2 + b_3 x^3 = (1 + c_1 x + c_2 x^2 + c_3 x^3) \times$$
$$\times (a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6).$$

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

$$b_0 + b_1 x + b_2 x^2 + b_3 x^3 = a_0 + x(a_1 + a_0 c_1) +$$
$$+ x^2(a_2 + a_1 c_1 + a_0 c_2) + x^3(a_3 + a_2 c_1 + a_1 c_2 + a_0 c_3) +$$
$$+ x^4(a_4 + a_3 c_1 + a_2 c_2 + a_1 c_3) + x^5(a_5 + a_4 c_1 + a_3 c_2 + a_2 c_3) +$$
$$+ x^6(a_6 + a_5 c_1 + a_4 c_2 + a_3 c_3).$$

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

$$\begin{aligned} b_0 &= a_0, \\ b_1 &= a_1 + a_0 c_1, \\ b_2 &= a_2 + a_1 c_1 + a_0 c_2, \\ b_3 &= a_3 + a_2 c_1 + a_1 c_2 + a_0 c_3, \\ 0 &= a_4 + a_3 c_1 + a_2 c_2 + a_1 c_3, \\ 0 &= a_5 + a_4 c_1 + a_3 c_2 + a_2 c_3, \\ 0 &= a_6 + a_5 c_1 + a_4 c_2 + a_3 c_3. \end{aligned} \qquad (2.19)$$

Решив эту систему, найдем коэффициенты b_0, b_1, b_2, b_3, c_1, c_2, c_3, необходимые для аппроксимации (2.17).

Пример. Рассмотрим рациональное приближение для функции $f(x) = \sin(\pi x/2)$. Воспользуемся представ-

лением (2.17), 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 (2.19), причем значения коэффициентов a_0, a_1, ..., a_6 разложения функции в ряд Тейлора (2.18) можем взять из выражения (2.10), т. е.

$$a_0 = 0, \quad a_1 = \frac{\pi}{2}, \quad a_2 = 0, \quad a_3 = -\frac{\pi^3}{8 \cdot 3!},$$

$$a_4 = 0, \quad a_5 = \frac{\pi^5}{32 \cdot 5!}, \quad a_6 = 0.$$

Система уравнений (2.19) в данном случае примет вид

$$b_1 = \frac{\pi}{2},$$

$$b_3 = -\frac{\pi^3}{8 \cdot 3!} + \frac{\pi}{2} c_2,$$

$$0 = \frac{\pi^5}{32 \cdot 5!} - \frac{\pi^3}{8 \cdot 3!} c_2.$$

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

Таким образом, дробно-рациональное приближение (2.17) для функции $\sin(\pi x/2)$ примет вид

$$\sin\frac{\pi x}{2} = \frac{(\pi/2)\, x - (7\pi^3/480)\, x^3}{1 + (\pi^2/80)\, x^2}, \qquad (2.20)$$

Это приближение по точности равносильно аппроксимации (2.10) с учетом членов до пятого порядка включительно.

На практике с целью экономии числа операций выражение (2.17) представляется в виде *цепной дроби*. Представим в таком виде дробно-рациональное выражение (2.20). 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

$$\sin\frac{\pi x}{2} = k_1\left(x + \frac{k_2}{x + \dfrac{k_3}{x}}\right). \tag{2.21}$$

Для вычисления значения функции по этой формуле требуется намного меньше операций (два деления, два сложения, одно умножение), чем для вычисления с помощью выражения (2.20) или усеченного ряда Тейлора (2.10) (даже с использованием правила Горнера).

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

$$e^x = 1 + \cfrac{x}{-\dfrac{x}{2} + \dfrac{k_0 + k_1 x^2}{1 + k_2 x^2}}, \tag{2.22}$$

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

$$k_2 = 0.0166411490538, \quad -\frac{1}{2}\ln 2 \leqslant x \leqslant \frac{1}{2}\ln 2, \quad |\Delta| \leqslant 10^{-10};$$

$$\ln(1+x) = k_0 + \cfrac{x}{k_1 + \cfrac{x}{k_2 + \cfrac{x}{k_3 + \cfrac{x}{k_4 + \dfrac{x}{k_5}}}}}, \tag{2.23}$$

$$k_0 = 0.0000000894, \quad k_1 = 1.0000091365,$$
$$k_2 = 2.0005859000, \quad k_3 = 3.0311932666,$$
$$k_4 = 1.0787748225, \quad k_5 = 8.8952784060,$$
$$0 \leqslant x \leqslant 1, \quad |\Delta| \leqslant 10^{-7};$$

$$\operatorname{tg}\frac{\pi x}{4} = x\left(k_0 + \cfrac{x^2}{k_1 + \cfrac{x^2}{k_2 + \dfrac{x^2}{k_3}}}\right), \tag{2.24}$$

$$k_0 = 0.7853980289, \quad k_1 = 6.1922344479,$$
$$k_2 = -0.6545887679, \quad k_3 = 491.0013934779,$$
$$-1 \leqslant x \leqslant 1, \quad |\Delta| \leqslant 2 \cdot 10^{-8};$$

$$\operatorname{arctg} x = x \left\{ k_0 + \cfrac{x^2}{k_1 + \cfrac{x^2}{k_2 + \cfrac{x^2}{k_3 + \cfrac{x^2}{k_4}}}} \right\}, \tag{2.25}$$

$$k_0 = 0.99999752, \quad k_1 = -3.00064286,$$
$$k_2 = -0.55703890, \quad k_3 = -17.03715998,$$
$$k_4 = -0.20556880, \quad -1 \leqslant x \leqslant 1, \quad |\Delta| \leqslant 2 \cdot 10^{-7}.$$

§ 3. Интерполирование

1. Линейная и квадратичная интерполяции. Простейшим и часто используемым видом локальной интерполяции является *линейная 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

$$y = a_i x + b_i, \quad x_{i-1} \leqslant x \leqslant x_i, \tag{2.26}$$
$$a_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}, \quad b_i = y_{i-1} - a_i x_{i-1}.$$

Следовательно, при использовании линейной интерполяции сначала нужно определить интервал, в который попадает значение аргумента x, а затем подставить его

в формулу (2.26) и найти приближенное значение функции в этой точке. Блок-схема данного алгоритма представлена на рис. 8. Попытайтесь разобраться, будет ли работать алгоритм по этой блок-схеме, если окажется, что $x < x_0$ или $x > x_n$.

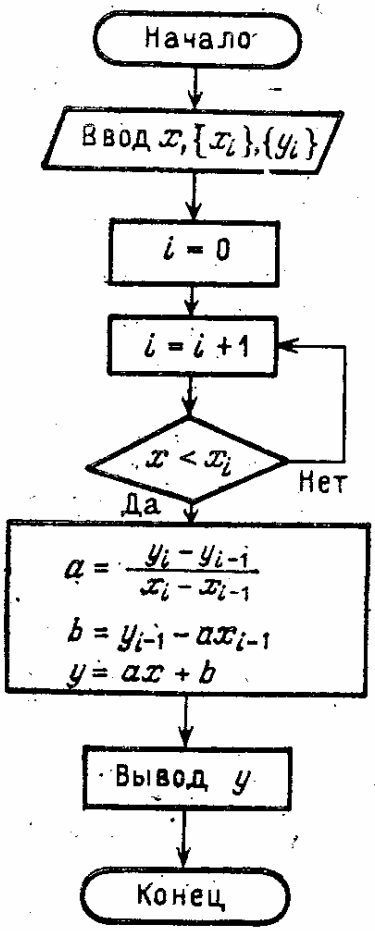


Рис. 8. Блок-схема линейной интерполяции

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

$$y = a_i x^2 + b_i x + c_i, \quad x_{i-1} \leqslant x \leqslant x_{i+1}, \tag{2.27}$$

содержит три неизвестных коэффициента a_i, b_i, c_i, для определения которых необходимы три уравнения. Ими служат условия прохождения параболы (2.27) 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

$$\begin{aligned} a_i x_{i-1}^2 + b_i x_{i-1} + c_i &= y_{i-1}, \\ a_i x_i^2 + b_i x_i + c_i &= y_i, \\ a_i x_{i+1}^2 + b_i x_{i+1} + c_i &= y_{i+1}. \end{aligned} \tag{2.28}$$

Алгоритм вычисления приближенного значения функции с помощью квадратичной интерполяции можно представить в виде блок-схемы, как и для случая линейной интерполяции (см. рис. 8). Вместо формулы (2.26) нужно использовать (2.27) с учетом решения системы линейных уравнений (2.28). 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 (2.26). Значение $x = 0.32$ находится между узлами $x_{i-1} = 0.30$ и $x_i = 0.40$. В этом случае

$$a_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} = \frac{5.07 - 3.63}{0.40 - 0.30} = 14.4,$$

$$b_i = y_{i-1} - a_i x_{i-1} = 3.63 - 14.4 \cdot 0.30 = -0.69,$$

$$y \approx 14.4x - 0.69 = 14.4 \cdot 0.32 - 0.69 = 3.92.$$

Найдем теперь приближенное значение функции с помощью формулы квадратичной интерполяции (2.27). Составим систему уравнений (2.28) 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 (2.28) 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 $y \approx 18.67 \cdot 0.32^2 + 1.33 \cdot 0.32 + 1.55 = 3.89$.

2. 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. 9), то между точками закрепления этот стержень (механический сплайн) примет некоторую форму, минимизирующую его потенциальную энергию.

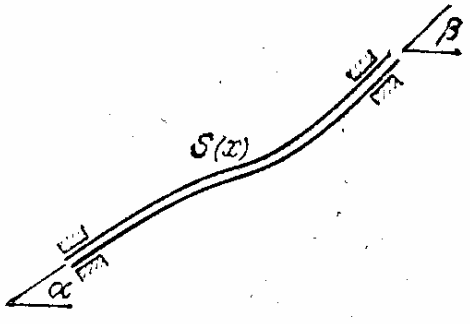


Рис. 9

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

третьей степени. Запишем ее в виде

$$S(x) = a_i + b(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \quad x_{i-1} \leqslant x \leqslant x_i. \tag{2.29}$$

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. $S(x_{i-1}) = y_{i-1}$, $S(x_i) = y_i$. Эти условия можно записать в виде

$$S(x_{i-1}) = a_i = y_{i-1}, \tag{2.30}$$

$$S(x_i) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, \quad h_i = x_i - x_{i-1}, \quad i = 1, 2, \ldots, n. \tag{2.31}$$

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

Вычислим производные многочлена (2.29):

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

Приравнивая в каждом внутреннем узле $x = x_i$ значения этих производных, вычисленные в левом и правом от узла интервалах, получаем $2n - 2$ уравнений

$$b_{i+1} = b_i + 2h_i c_i + 3h_i^2 d_i, \tag{2.32}$$

$$c_{i+1} = c_i + 3h_i d_i, \quad i = 1, 2, \ldots, n-1. \tag{2.33}$$

Недостающие два соотношения получаются из условий закрепления концов сплайна.

В частности, при свободном закреплении концов (см. рис. 9) можно приравнять нулю кривизну линии в этих точках. Такая функция, называемая *свободным кубическим сплайном*, обладает свойством минимальной кривизны, т. е. она самая гладкая среди всех интерполяционных функций данного класса. Из условий нулевой кривизны на концах следуют равенства нулю вторых производных в этих точках:

$$S''(x_0) = c_1 = 0, \quad S''(x_n) = 2c_n + 6d_n h_n = 0. \tag{2.34}$$

Уравнения (2.30)—(2.34) составляют систему линейных алгебраических уравнений для определения $4n$ ко-

эффициентов a_i, b_i, c_i, d_i $(i = 1, 2, \ldots, n)$. Ее можно решить одним из методов, изложенных в гл. 4.

Однако с целью экономии памяти ЭВМ и машинного времени эту систему можно привести к более удобному виду. Из условия (2.30) сразу можно найти все коэффициенты a_i. Далее из (2.33), (2.34) получим

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad i = 1, 2, \ldots, n-1, \quad d_n = -\frac{c_n}{3h_n}. \tag{2.35}$$

Подставим эти соотношения, а также значения $a_i = y_i - 1$ в (2.31) и найдем отсюда коэффициенты

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i), \quad i = 1, 2, \ldots, n-1,$$
$$b_n = \frac{y_n - y_{n-1}}{h_n} - \frac{2}{3}h_n c_n. \tag{2.36}$$

Учитывая выражения (2.35) и (2.36), исключаем из уравнения (2.32) коэффициенты d_i и b_i. Окончательно получим следующую систему уравнений только для коэффициентов c_i:

$$c_1 = 0, \quad c_{n+1} = 0,$$
$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}}\right), \quad i = 2, 3, \ldots, n. \tag{2.37}$$

Матрица этой системы 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 (2.37) коэффициентам c_i легко вычислить коэффициенты d_i, b_i.

3. Многочлен Лагранжа. Перейдем к случаю глобальной интерполяции, т. е. построению интерполяционного многочлена, единого для всего отрезка $[x_0, x_n]$. При этом, естественно, график интерполяционного многочлена должен проходить через все заданные точки.

Запишем искомый многочлен в виде

$$\varphi(x) = a_0 + a_1 x + \ldots + a_n x^n. \tag{2.38}$$

Из условий равенства значений этого многочлена в узлах x_i соответствующим заданным табличным значениям y_i получим следующую систему уравнений для нахожде-

ния коэффициентов $a_0, a_1, \ldots, a_n$:

$$\begin{aligned} a_0 + a_1 x_0 + \ldots + a_n x_0^n &= y_0, \\ a_0 + a_1 x_1 + \ldots + a_n x_1^n &= y_1, \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ a_0 + a_1 x_n + \ldots + a_n x_n^n &= y_n. \end{aligned} \tag{2.39}$$

Можно показать, что эта система имеет единственное решение, если среди узлов интерполяции нет совпадающих, т. е. если $x_i \neq x_j$ при $i \neq j$. Решив эту систему, найдем коэффициенты интерполяционного многочлена (2.38). Заметим вместе с тем, что такой путь построения интерполяционного многочлена требует значительного объема вычислений, особенно при большом числе узлов. Существуют более простые алгоритмы построения интерполяционных многочленов.

Будем искать многочлен в виде линейной комбинации многочленов степени n:

$$L(x) = y_0 l_0(x) + y_1 l_1(x) + \ldots + y_n l_n(x). \tag{2.40}$$

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 отвечает многочлен вида

$$l_0(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_n)}{(x_0 - x_1)(x_0 - x_2)\ldots(x_0 - x_n)}. \tag{2.41}$$

Действительно, $l_0(x_0) = 1$ при $x = x_0$. При $x = x_1, x_2, \ldots, x_n$ числитель выражения (2.41) обращается в нуль. По аналогии с (2.41) получим

$$\begin{aligned} l_1(x) &= \frac{(x - x_0)(x - x_2)\ldots(x - x_n)}{(x_1 - x_0)(x_1 - x_2)\ldots(x_1 - x_n)}, \\ l_2(x) &= \frac{(x - x_0)(x - x_1)(x - x_3)\ldots(x - x_n)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)\ldots(x_2 - x_n)}, \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ l_i(x) &= \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}, \\ \cdots\cdots\cdots\cdots\cdots\cdots & \end{aligned} \tag{2.42}$$

Подставляя в (2.40) выражения (2.41), (2.42), находим

$$L(x) = \sum_{i=0}^{n} y_i \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}. \tag{2.43}$$

Эта формула называется *интерполяционным многочленом Лагранжа.*

Покажем, что этот многочлен является единственным. Допустим противоположное: пусть существует еще один многочлен $F(x)$ степени n, принимающий в узлах интерполяции заданные значения, т. е. $F(x_i)=y_i$. Тогда разность $R(x)=L(x)-F(x)$, являющаяся многочленом степени n (или ниже), в узлах x_i равна

$$R(x_i)=L(x_i)-F(x_i)=0, \quad i=0, 1, \ldots, n.$$

Это означает, что многочлен $R(x)$ степени не больше n имеет $n+1$ корней. Отсюда следует, что $R(x)\equiv 0$ и $F(x)=L(x)$.

Из формулы (2.43) можно получить выражения для линейной ($n=1$) и квадратичной ($n=2$) интерполяций:

$$L(x)=\frac{x-x_1}{x_0-x_1}y_0+\frac{x-x_0}{x_1-x_0}y_1, \quad n=1;$$

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

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 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.

4. 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:

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

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

Можно составить *вторые разности* функции:

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

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

$$\Delta^k y_0 = y_k - k y_{k-1} + \frac{k(k-1)}{2!} y_{k-2} + \ldots + (-1)^k y_0. \quad (2.44)$$

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:

$$y_k = y_0 + k\Delta y_0 + \frac{k(k-1)}{2!} \Delta^2 y_0 + \ldots + \Delta^k y_0. \quad (2.45)$$

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

$$N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots$$
$$\ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \quad (2.46)$$

График многочлена должен проходить через заданные узлы, т. е. $N(x_i) = y_i$ $(i = 0, 1, \ldots, n)$. Эти условия ис-

пользуем для нахождения коэффициентов многочлена:

$$N(x_0) = a_0 = y_0,$$
$$N(x_1) = a_0 + a_1(x_1 - x_0) = a_0 + a_1 h = y_1,$$
$$N(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) =$$
$$= a_0 + 2a_1 h + 2a_2 h^2 = y_2,$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$

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 (2.46), получаем следующий вид *интерполяционного многочлена Ньютона*:

$$N(x) = y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2! h^2}(x - x_0)(x - x_1) + \ldots$$

$$\ldots + \frac{\Delta^n y_0}{n! h^n}(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \quad (2.47)$$

Конечные разности $\Delta^k y_0$ могут быть вычислены по формуле (2.44).

Формулу (2.47) часто записывают в другом виде. Для этого вводится переменная $t = (x - x_0)/h$; тогда

$$x = x_0 + th, \quad \frac{x - x_1}{h} = \frac{x - x_0 - h}{h} = t - 1,$$

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

С учетом этих соотношений формулу (2.47) можно переписать в виде

$$N(x_0 + th) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \ldots$$

$$\ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_0. \quad (2.48)$$

Полученное выражение может аппроксимировать данную функцию $y = f(x)$ на всем отрезке изменения аргумента $[x_0, x_n]$. Однако более целесообразно (с точки зрения повышения точности расчетов и уменьшения числа членов в (2.48)) ограничиться случаем $t < 1$, т. е. использовать формулу (2.48) для $x_0 \leqslant x \leqslant x_1$. Для других значений аргумента, например для $x_1 \leqslant x \leqslant x_2$, вместо x_0 лучше взять значение x_1. Таким образом, интерполяционный многочлен Ньютона можно записать в виде

$$N(x_i + th) = y_i + t\Delta y_i + \frac{t(t-1)}{2!}\Delta^2 y_i + \ldots$$

$$\ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n y_i, \quad i = 0, 1, \ldots \quad (2.49)$$

Полученное выражение называется *первым интерполяционным многочленом Ньютона для интерполирования вперед.*

Интерполяционную формулу (2.49) обычно используют для вычисления значений функции в точках левой половины рассматриваемого отрезка. Это объясняется следующим. Разности $\Delta^k y_i$ вычисляются через значения функции $y_i, y_{i+1}, \ldots, y_{i+k}$, причем $i + k \leqslant n$; поэтому при больших значениях i мы не можем вычислить разности высших порядков $(k \leqslant n - i)$. Например, при $i = n - 3$ в (2.49) можно учесть только Δy, $\Delta^2 y$ и $\Delta^3 y$.

Для правой половины рассматриваемого отрезка разности лучше вычислять справа налево. В этом случае

$$t = (x - x_n)/h, \quad (2.50)$$

т. е. $t < 0$, и интерполяционный многочлен Ньютона можно получить в виде

$$N(x_n + th) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \ldots$$

$$\ldots + \frac{t(t+1)\ldots(t+n-1)}{n!}\Delta^n y_0. \quad (2.51)$$

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

Рассмотрим пример применения интерполяционной формулы Ньютона при ручном счете.

Пример. Вычислить в точках $x = 0.1$, 0.9 значения функции $y = f(x)$, заданной табл. 1.

Процесс вычислений удобно свести в ту же табл. 1. Каждая последующая конечная разность получается путем вычитания в предыдущей колонке верхней строки

Таблица 1

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
0	1.2715	1.1937	—0.0146	0.0007	—0.0001	0.0000
0.2	2.4652	1.1791	—0.0139	0.0006	—0.0001	
0.4	3.6443	1.1652	—0.0133	0.0005		
0.6	4.8095	1.1519	—0.0128			
0.8	5.9614	1.1391				
1.0	7.1005					

из нижней. При $x = 0.1$ имеем $t = (x - x_0)/h = (0.1 - 0)/0.2 = 0.5$. По формуле (2.48) получим

$$f(0.1) \approx N(0.1) = 1.2715 + 0.5 \cdot 1.1937 +$$
$$+ \frac{0.5\,(0.5-1)}{2!} \cdot (-0.0146) + \frac{0.5\,(0.5-1)\,(0.5-2)}{3!} \cdot 0.0007 +$$
$$+ \frac{0.5(0.5-1)\,(0.5-2)\,(0.5-3)}{4!} \cdot (-0.0001) =$$
$$= 1.2715 + 0.59685 + 0.00182 + 0.00004 + 0.000004 =$$
$$= 1.8702.$$

Для сравнения по формуле линейной интерполяции получаем

$$f(0.1) \approx 1.8684.$$

Значение функции в точке $x = 0.9$ нужно вычислять по формуле (2.51). В этом случае имеем $t = (x - x_n)/h = (0.9 - 1)/0.2 = -0.5$. Тогда

$$f(0.9) \approx N(0.9) = 7.1005 - 0.5 \cdot 1.1391 -$$
$$- \frac{0.5\,(-0.5+1)}{2!} \cdot (-0.0128) -$$
$$- \frac{0.5\,(-0.5+1)\,(-0.5+2)}{3!} \cdot 0.0005 -$$
$$- \frac{0.5\,(-0.5+1)\,(-0.5+2)\,(-0.5+3)}{4!} \cdot (-0.0001) =$$
$$= 7.1005 - 0.5696 + 0.0016 - 0.00003 + 0.000004 = 6.5325.$$

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).

5. Точность интерполяции. График интерполяционного многочлена $y = F(x)$ 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 тождественное совпадение: $f(x) = F(x)$. В общем случае в точках, отличных от узлов интерполяции, $R(x) = f(x) - F(x) \neq 0$. 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

$$R_L(x) = \frac{(x - x_0)(x - x_1) \ldots (x - x_n)}{(n+1)!} f^{(n+1)}(x_*). \quad (2.52)$$

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

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

то можно записать формулу для оценки остаточного члена:

$$|R_L(x)| \leqslant \frac{(x - x_0)(x - x_1) \ldots (x - x_n)}{(n+1)!} M_{n+1}.$$

Остаточный член интерполяционного многочлена Ньютона можно записать в виде

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

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

$$R_N(x) = \frac{t(t-1)\ldots(t-n)}{(n+1)!}\Delta^{n+1}y_0. \qquad (2.53)$$

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.

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).

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 их производные 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:

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

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_j), $(i, j = 0, 1, 2)$. 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 формуле 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

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

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

$$z = \frac{1}{D_3}(D_0 - D_1 x - D_2 y), \tag{2.54}$$

$$D_0 = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix}, \quad D_1 = \begin{vmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{vmatrix},$$
$$D_2 = \begin{vmatrix} x_1 & z_1 & 1 \\ x_2 & z_2 & 1 \\ x_3 & z_3 & 1 \end{vmatrix}, \quad D_3 = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}.$$

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 (2.54). Вычислим значения определителей

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

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

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 временные 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

$$y = f(x), \tag{2.55}$$

значения которой при $x = x_i$ $(i = 0, 1, \ldots, n)$ мало отличаются от опытных данных y_i. Приближенная функциональная зависимость (2.55), 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, а 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

$$y = ax + b. \tag{2.56}$$

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 (2.56). 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:

$$\xi = \varphi(x, y), \quad \eta = \psi(x, y). \tag{2.57}$$

Функции $\varphi(x, y)$ и $\psi(x, y)$ 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 (2.57) 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

$$y = ax^2 + bx + c. \qquad (2.58)$$

Возможность использования этой формулы для случая равноотстоящих точек x_i можно проверить путем вычисления вторых разностей $\Delta^2 y_i = y_{i+1} - 2y_i + y_{i-1}$. При $\Delta^2 y_i \approx \approx \text{const}$ в качестве эмпирической формулы может быть выбрана (2.58).

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

$$y = \varphi(x, a_0, a_1, \ldots, a_m), \qquad (2.59)$$

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 (2.59) в точках x_i. Разность между этими значениями (отклонения) обозначим через ε_i. Тогда

$$\varepsilon_i = \varphi(x_i, a_0, a_1, \ldots, a_m) - y_i, \quad i = 0, 1, \ldots, n. \qquad (2.60)$$

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 (2.59) через выбранные точки:

$$\varphi(x_j^0, a_0, a_1, \ldots, a_m) = y_j^0, \quad j = 0, 1, \ldots, m, \qquad (2.61)$$

Из этой системы уравнений находим 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_1^0, y_1^0) и (x_2^0, y_2^0), и уравнения (2.61) примут вид

$$\begin{aligned} ax_1^0 + b &= y_1^0, \\ ax_2^0 + b &= y_2^0. \end{aligned} \tag{2.62}$$

Можно также записать уравнение прямой, проходящей через эти выбранные точки. В этом случае не нужно решать систему (2.62).

Рассмотрим еще один способ определения параметров эмпирической формулы — *метод средних*. Он состоит в том, что параметры $a_0, a_1, \ldots, a_m$ зависимости (2.59) определяются с использованием условия равенства нулю суммы отклонений (2.60) во всех точках x_i:

$$\sum_{i=0}^{n} \varepsilon_i = \sum_{i=1}^{n} [\varphi(x_i, a_0, a_1, \ldots, a_m) - y_i] = 0. \tag{2.63}$$

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 (2.63) путем группировки отклонений ε_i разбивается на систему, состоящую из $m+1$ уравнений. Например,

$$\begin{aligned} \varepsilon_0 + \varepsilon_1 + \varepsilon_2 &= 0, \\ \varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6 &= 0, \\ &\cdots\cdots\cdots \\ \varepsilon_{n-1} + \varepsilon_n &= 0. \end{aligned} \tag{2.64}$$

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, 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

$$x = At^2 + Bt. \qquad (2.65)$$

Для определения параметров A, B нужно получить два уравнения. Воспользуемся методом средних и запишем уравнение (2.63) 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 (2.65) и табличные данные, получаем

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

Или окончательно

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

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

Следовательно, эмпирическую формулу (2.65), 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: $a = 2A = -0{,}60$ м/с^2, $v_0 = B = 39.07$ 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 (2.60) для всех точек $x_0, x_1, \ldots, x_n$:

$$S = \sum_{i=0}^{n} \varepsilon_i^2 = \sum_{i=0}^{n} [\varphi(x_i, a_0, a_1, \ldots, a_m) - y_i]^2. \quad (2.66)$$

Параметры $a_0, a_1, \ldots, a_m$ эмпирической формулы (2.59) 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:

$$\frac{\partial S}{\partial a_0} = 0, \quad \frac{\partial S}{\partial a_1} = 0, \ldots, \frac{\partial S}{\partial a_m} = 0. \quad (2.67)$$

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

Рассмотрим применение метода наименьших квадратов для частного случая, широко используемого на практике. В качестве эмпирической функции рассмотрим многочлен

$$\varphi(x) = a_0 + a_1 x + \ldots + a_m x^m. \quad (2.68)$$

Формула (2.66) для определения суммы квадратов отклонений S примет вид

$$S = \sum_{i=0}^{n} (a_0 + a_1 x_i + \ldots + a_m x_i^m - y_i)^2. \quad (2.69)$$

Для составления системы уравнений (2.67) найдем частные производные функции $S = S(a_0, a_1, \ldots, a_m)$:

$$\frac{\partial S}{\partial a_0} = 2\sum_{i=0}^{n}(a_0 + a_1x_i + \ldots + a_mx_i^m - y_i),$$

$$\frac{\partial S}{\partial a_1} = 2\sum_{i=0}^{n}(a_0 + a_1x_i + \ldots + a_mx_i^m - y_i)x_i,$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\frac{\partial S}{\partial a_m} = 2\sum_{i=0}^{n}(a_0 + a_1x_i + \ldots + a_mx_i^m - y_i)x_i^m.$$

Приравнивая эти выражения нулю в соответствии с уравнениями (2.67) и собирая коэффициенты при неизвестных $a_0, a_1, \ldots, a_m$, получаем следующую систему уравнений:

$$\begin{aligned}
&(n+1)a_0 + a_1\sum_{i=0}^{n}x_i + a_2\sum_{i=0}^{n}x_i^2 + \ldots + a_m\sum_{i=0}^{n}x_i^m = \sum_{i=0}^{n}y_i,\\
&a_0\sum_{i=0}^{n}x_i + a_1\sum_{i=0}^{n}x_i^2 + a_2\sum_{i=0}^{n}x_i^3 + \ldots + a_m\sum_{i=0}^{n}x_i^{m+1} = \sum_{i=0}^{n}x_iy_i,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_0\sum_{i=0}^{n}x_i^m + a_1\sum_{i=0}^{n}x_i^{m+1} + a_2\sum_{i=0}^{n}x_i^{m+2} + \ldots + a_m\sum_{i=0}^{n}x_i^{2m} = \sum_{i=0}^{n}x_i^my_i.
\end{aligned} \tag{2.70}$$

Решая эту систему линейных уравнений, получаем коэффициенты $a_0, a_1, \ldots, a_m$ многочлена (2.68), которые являются искомыми параметрами эмпирической формулы.

Систему (2.70) можно записать в более компактном виде:

$$\begin{aligned}
b_{00}a_0 + b_{01}a_1 + \ldots + b_{0m}a_m &= c_0,\\
b_{10}a_0 + b_{11}a_1 + \ldots + b_{1m}a_m &= c_1,\\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots&\\
b_{m0}a_0 + b_{m1}a_1 + \ldots + b_{mm}a_m &= c_m;
\end{aligned} \tag{2.71}$$

$$b_{kl} = \sum_{i=0}^{n}x_i^{k+l}, \quad c_k = \sum_{i=0}^{n}x_i^ky_i, \quad k, l = 0, 1, \ldots, m. \tag{2.72}$$

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

Решение. Если изобразить данные табличные значения на графике (рис. 10), то легко убедиться, что в качестве эмпирической формулы для аппроксимации функции $y = f(x)$ можно принять параболу, т. е. квадратный трехчлен:

$$y \approx \varphi(x) = a_0 + a_1 x + a_2 x^2. \quad (2.73)$$

В данном случае имеем $m = 2$, $n = 4$, и система уравнений (2.71) примет вид

$$\begin{aligned} b_{00}a_0 + b_{01}a_1 + b_{02}a_2 &= c_0, \\ b_{10}a_0 + b_{11}a_1 + b_{12}a_2 &= c_1, \\ b_{20}a_0 + b_{21}a_1 + b_{22}a_2 &= c_2. \end{aligned} \quad (2.74)$$

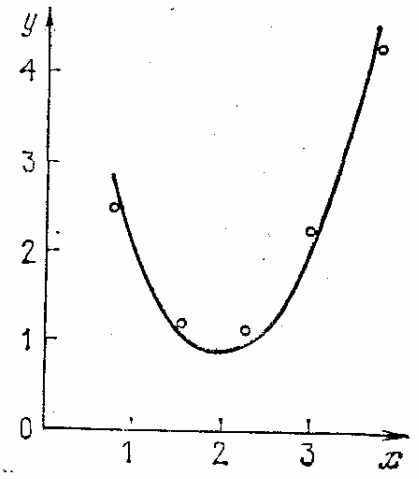


Рис. 10

Коэффициенты этой системы могут быть вычислены по формулам (2.72), где $i = 0, 1, 2, 3, 4$:

$$b_{00} = 5,$$

$$b_{01} = \sum x_i = 0.75 + 1.50 + 2.25 + 3.00 + 3.75 = 11.25,$$

$$b_{02} = \sum x_i^2 = 0.75^2 + 1.50^2 + 2.25^2 + 3.00^2 + 3.75^2 = 30.94,$$

$$b_{10} = \sum x_i = 11.25, \quad b_{11} = \sum x_i^2 = 30.94,$$

$$b_{12} = \sum x_i^3 = 94.92,$$

$$b_{20} = \sum x_i^2 = 30.94, \quad b_{21} = \sum x_i^3 = 94.92,$$

$$b_{22} = \sum x_i^4 = 309.76,$$

$$c_0 = \sum y_i = 2.50 + 1.20 + 1.12 + 2.25 + 4.28 = 11.35,$$

$$c_1 = \sum x_i y_i = 29.00, \quad c_2 = \sum x_i^2 y_i = 90.21.$$

Система уравнений (2.74) запишется в виде

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

Отсюда находим значения параметров эмпирической формулы: $a_0 = 5.54$, $a_1 = -4.73$, $a_2 = 1.19$. Таким образом, получаем следующую аппроксимацию функции, заданной в табличном виде:

$$y \approx 5.54 - 4.73x + 1.19x^2. \qquad (2.75)$$

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

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

Результаты вычислений представим в виде таблицы

x	$\varphi(x)$	y	ε	δy
0.75	2.66	2.50	0.16	0.064
1.50	1.12	1.20	—0.08	—0.067
2.25	0.92	1.12	—0.20	—0.179
3.00	2.06	2.25	—0.19	—0.084
3.75	4.54	4.28	0.26	0.061

На рис. 10 построен график найденной эмпирической формулы. Точками, как уже отмечалось, нанесены заданные табличные значения функции.

Замечание. Как видно из рассмотренного примера, некоторые коэффициенты системы (2.74) равны: $b_{01} = b_{10}$, $b_{02} = b_{11} = b_{20}$, $b_{12} = b_{21}$. Из формул (2.72) нетрудно увидеть, что равны все коэффициенты b_{kl} при $k + l = \text{const}$.

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 $y_0, y_1, \ldots, y_n$ в точках $x_0, x_1, \ldots, x_n$. Значения аргумента x_i предполагаются равноотстоящими, а опытные данные y_i — имеющими одинаковую точность. Предполагается также, что функция $y = f(x)$ на произ-

вольной 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 значений 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. Составить блок-схему алгоритмов 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 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. Составить блок-схему алгоритма вычисления функции с помощью квадратичной интерполяции.

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

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

9. 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

10. 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 и составить блок-схему вычисления параметра E.

11. 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.

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:

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

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

$$y' \approx \Delta y / \Delta x. \quad (3.2)$$

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_0, y_1, ... при $x = x_0$, x_1, ... Пусть *шаг* — разность между соседними значениями аргумента — постоянный и равен h. Запишем выражения для производной y_1 при $x = x_1$. В зависимости от способа вычисления конечных разностей получаем разные формулы для вычисления производной в одной и той же точке:

$$\Delta y_1 = y_1 - y_0, \quad \Delta x = h, \quad y_1' \approx \frac{y_1 - y_0}{h} \quad (3.3)$$

с помощью *левых разностей*;

$$\Delta y_1 = y_2 - y_1, \quad \Delta x = h, \quad y_1' \approx \frac{y_2 - y_1}{h} \tag{3.4}$$

с помощью *правых разностей*;

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

с помощью *центральных разностей.*

Можно найти также выражения для старших производных. Например,

$$y_1'' = (y_1')' \approx \frac{y_2' - y_1'}{h} \approx \frac{(y_2 - y_1)/h - (y_1 - y_0)/h}{h} =$$

$$= \frac{y_2 - 2y_1 + y_0}{h^2}. \tag{3.6}$$

Таким образом, по формуле (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 $f'(x)$, $f''(x)$, ...:

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

В качестве приближенного значения производной порядка k функции $f(x)$ можно принять соответствующее значение 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 $O(h^k)$. Показатель степени k называется *порядком* погрешности аппроксимации производной (или просто порядком аппроксимации). При этом предполагается, что значение шага по модулю меньше единицы.

Оценку погрешности легко проиллюстрировать с помощью ряда Тейлора

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

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

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

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 $(k = 1)$. 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

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

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):

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

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, и сведения по этому вопросу можно найти в более полных курсах по численным методам и теории разностных схем. 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 ЭВМ. В отличие от погрешности аппроксимации погрешность округления возрастает с уменьшением шага 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.

Оптимальная точность может быть достигнута 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 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 — 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 по-

стоянным шагом $h = x_i - x_{i-1}$ $(i = 1, 2, \ldots, n)$, может быть аппроксимирована интерполяционным многочленом Ньютона (2.3):

$$y \approx N(x_0 + th) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \ldots$$

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

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:

$$y' \approx \frac{1}{h}\left(\Delta y_0 + \frac{2t-1}{2!}\Delta^2 y_0 + \frac{3t^2 - 6t + 2}{3!}\Delta^3 y_0 + \right.$$

$$+ \frac{4t^3 - 18t^2 + 22t - 6}{4!}\Delta^4 y_0 +$$

$$\left. + \frac{5t^4 - 40t^3 + 105t^2 - 100t + 24}{5!}\Delta^5 y_0 + \ldots\right),$$

$$y'' \approx \frac{1}{h^2}\left(\Delta^2 y_0 + \frac{6t-6}{3!}\Delta^3 y_0 + \frac{12t^2 - 36t + 22}{4!}\Delta^4 y_0 + \right.$$

$$\left. + \frac{20t^3 - 120t^2 + 210t - 100}{5!}\Delta^5 y_0 + \ldots\right),$$

. . . . . .

Пример. Вычислить в точке $x = 0.1$ первую и вторую производные функции, заданной таблицей (табл. 2).

Здесь $h = 0.1$, $t = (0.1 - 0)/0.1 = 1$. Используя полученные выше формулы, находим

$$y' \approx 10\left(0.5274 + \frac{2\cdot 1 - 1}{2}\cdot 0.0325 + \frac{3\cdot 1 - 6\cdot 1 + 2}{6}\cdot 0.0047 + \right.$$

$$\left. + \frac{4\cdot 1 - 18\cdot 1 + 22\cdot 1 - 6}{24}\cdot 0.0002\right) = 5{,}436,$$

$$y'' \approx 100\left(0{,}0325 + \frac{6\cdot 1 - 6}{6}\cdot 0.0047 + \right.$$

$$\left. + \frac{12 - 36 + 22}{24}\cdot 0.0002\right) = 3{,}25.$$

Интерполяционные многочлены Ньютона (а также Стирлинга и Бесселя) дают выражения для производных через разности $\Delta^k y$ $(k = 1, 2, \ldots)$. 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 $(x_i - x_{i-1} = h = \text{const}, i = 1, 2, \ldots, n)$.

Таблица 2

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

Запишем интерполяционный многочлен Лагранжа $L(x)$ и его остаточный член $R_L(x)$ (см. (2.43), (2.52)) для случая трех узлов интерполяции $(n = 2)$ и найдем их производные:

$$L(x) = \frac{1}{2h^2}[(x - x_1)(x - x_2)\, y_0 - 2(x - x_0)(x - x_2)\, y_1 + (x - x_0)(x - x_1)\, y_2],$$

$$R_L(x) = \frac{y'''_*}{3!}(x - x_0)(x - x_1)(x - x_2),$$

$$L'(x) = \frac{1}{2h^2}[(2x - x_1 - x_2)\, y_0 - 2(2x - x_0 - x_2)\, y_1 + (2x - x_0 - x_1)\, y_2],$$

$$R'_L(x) = \frac{y'''_*}{3!}[(x - x_1)(x - x_2) + (x - x_0)(x - x_2) + (x - x_0)(x - x_1)].$$

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

Запишем выражение для производной y_0' при $x = x_0$:

$$y_0' = L'(x_0) + R_L'(x_0) =$$

$$= \frac{1}{2h^2}[(2x_0 - x_1 - x_2)y_0 - 2(2x_0 - x_0 - x_2)y_1 +$$

$$+ (2x_0 - x_0 - x_1)y_2] + \frac{y_*'''}{3!}[(x_0 - x_1)(x_0 - x_2) +$$

$$+ (x_0 - x_0)(x_0 - x_2) + (x_0 - x_0)(x_0 - x_1)] =$$

$$= \frac{1}{2h}(-3y_0 + 4y_1 - y_2) + \frac{h^2}{3}y_*'''.$$

Аналогичные соотношения можно получить и для значений y_1', y_2' при $x = x_1$, x_2:

$$y_1' = \frac{1}{2h}(y_2 - y_0) - \frac{h^2}{6}y_*''', \quad y_2' = \frac{1}{2h}(y_0 - 4y_1 + 3y_2) + \frac{h^3}{3}y_*'''.$$

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

$$\begin{aligned}
y_0' &= \frac{1}{6h}(-11y_0 + 18y_1 - 9y_2 + 2y_3) - \frac{h^3}{4}y_*^{IV}, \\
y_1' &= \frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3) + \frac{h^3}{12}y_*^{IV}, \\
y_2' &= \frac{1}{6h}(y_0 - 6y_1 + 3y_2 + 2y_3) - \frac{h^3}{12}y_*^{IV}, \\
y_3' &= \frac{1}{6h}(-2y_0 + 9y_1 - 18y_2 + 11y_3) + \frac{h^3}{4}y_*^{IV}.
\end{aligned} \tag{3.9}$$

В случае пяти узлов ($n = 4$) получим

$$\begin{aligned}
y_0' &= \frac{1}{12h}(-25y_0 + 48y_1 - 36y_2 + 16y_3 - 3y_4) + \frac{h^4}{5}y_*^{V}, \\
y_1' &= \frac{1}{12h}(-3y_0 - 10y_1 + 18y_2 - 6y_3 + y_4) - \frac{h^4}{20}y_*^{V}, \\
y_2' &= \frac{1}{12h}(y_0 - 8y_1 + 8y_3 - y_4) + \frac{h^4}{30}y_*^{V}, \\
y_3' &= \frac{1}{12h}(-y_0 + 6y_1 - 18y_2 + 10y_3 + 3y_4) + \frac{h^4}{20}y_*^{V}, \\
y_4' &= \frac{1}{12h}(3y_0 - 16y_1 + 36y_2 - 48y_3 + 25y_4) + \frac{h^4}{5}y_*^{V}.
\end{aligned} \tag{3.10}$$

Таким образом, используя значения функции в $n+1$ узлах, 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:

$$\begin{aligned} y_i' &= \frac{1}{2h}(y_{i+1}-y_{i-1}) - \frac{h^2}{6}y_*''', && n=2,\\ y_i' &= \frac{1}{12h}(y_{i-2}-8y_{i-1}+8y_{i+1}-y_{i+2}) + \frac{h^4}{30}y_*^{\mathrm{V}}, && n=4. \end{aligned} \tag{3.11}$$

Они называются *аппроксимациями производных с помощью центральных разностей* и широко используются на практике.

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

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

$$\begin{aligned} y_0'' &= \frac{1}{h^2}(y_0-2y_1+y_2)+O(h),\\ y_1'' &= \frac{1}{h^2}(y_0-2y_1+y_2)+O(h^2),\\ y_2'' &= \frac{1}{h^2}(y_0-2y_1+y_2)+O(h). \end{aligned} \tag{3.12}$$

В случае четырех узлов ($n=3$) имеем

$$\begin{aligned} y_0'' &= \frac{1}{h^2}(2y_0-5y_1+4y_2-y_3)+O(h^2),\\ y_1'' &= \frac{1}{h^2}(y_0-2y_1+y_2)+O(h^2),\\ y_2'' &= \frac{1}{h^2}(y_1-2y_2+y_3)+O(h^2),\\ y_3'' &= \frac{1}{h^2}(-y_0+4y_1-5y_2+2y_3)+O(h^2). \end{aligned} \tag{3.13}$$

В случае пяти узлов ($n = 4$) имеем

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

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

$$y_2'' = \frac{1}{12h^2}(-y_0 + 16y_1 - 30y_2 + 16y_3 - y_4) + O(h^4), \quad (3.14)$$

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

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

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 $x_0, x_1, \ldots, x_n$:

$$y_i^{(k)} = c_0y_0 + c_1y_1 + \ldots + c_ny_n. \quad (3.15)$$

Предполагается, что эта формула имеет место для многочленов $y = 1$, $y = x - x_i$, $\ldots$, $y = (x - x_i)^n$. Подставляя последовательно эти выражения в (3.15), получаем систему $n + 1$ линейных алгебраических уравнений для определения неизвестных коэффициентов $c_0, c_1, \ldots, c_n$.

Пример. Найти выражение для производной y_1' в случае четырех равноотстоящих узлов ($n = 3$).

Равенство (3.15) запишется в виде

$$y_1' = c_0y_0 + c_1y_1 + c_2y_2 + c_3y_3. \quad (3.16)$$

Используем следующие многочлены:

$$y = 1, \; y = x - x_0, \; y = (x - x_0)^2, \; y = (x - x_0)^3. \quad (3.17)$$

Вычислим их производные:

$$y' = 0, \; y' = 1, \; y' = 2(x - x_0), \; y' = 3(x - x_0)^2. \quad (3.18)$$

Подставляем последовательно соотношения (3.17) и (3.18) соответственно в правую и левую части равенства (3.16) при $x = x_1$:

$$0 = c_0 \cdot 1 + c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 1,$$

$$1 = c_0(x_0 - x_0) + c_1(x_1 - x_0) + c_2(x_2 - x_0) + c_3(x_3 - x_0),$$

$$2(x_1 - x_0) = c_0(x_0 - x_0)^2 + c_1(x_1 - x_0)^2 + c_2(x_2 - x_0)^2 + c_3(x_3 - x_0)^2,$$

$$3(x_1 - x_0)^2 = c_0(x_0 - x_0)^3 + c_1(x_1 - x_0)^3 + c_2(x_2 - x_0)^3 + c_3(x_3 - x_0)^3.$$

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 равенство (3.16), находим выражение для производной:

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

5. Улучшение аппроксимации. Как видно из конечно-разностных соотношений для аппроксимаций производных (см. п. 3), порядок их точности прямо пропорционален числу узлов, используемых при аппроксимации. Однако с увеличением числа узлов эти соотношения становятся более 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

$$F(x) = f(x,\ h) + h^p\varphi(x) + O(h^{p+1}). \qquad (3.19)$$

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

$$F(x) = f(x,\ kh) + (kh)^p\varphi(x) + O((kh)^{p+1}). \qquad (3.20)$$

Приравнивая правые части равенств (3.19) и (3.20), 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 (3.19), получаем *формулу Рунге*:

$$F(x) = f(x,\ h) + \frac{f(x,\ h) - f(x,\ kh)}{k^p - 1} + O(h^{p+1}). \qquad (3.21)$$

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

$$f(x,\ h) = y'(1,\ 0.1) = \frac{y(1) - y(0.9)}{0.1} = \frac{1 - 0.729}{0.1} = 2.71,$$

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

По формуле Рунге найдем уточненное значение производной:

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

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

$$F(x) = \begin{vmatrix} f(x, h_1) & h_1^p & h_1^{p+1} & \ldots & h_1^{p+q-2} \\ f(x, h_2) & h_2^p & h_2^{p+1} & \ldots & h_2^{p+q-2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ f(x, h_q) & h_q^p & h_q^{p+1} & \ldots & h_q^{p+q-2} \end{vmatrix} \times$$

$$\times \begin{vmatrix} 1 & h_1^p & h_1^{p+1} & \ldots & h_1^{p+q-2} \\ 1 & h_2^p & h_2^{p+1} & \ldots & h_2^{p+q-2} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & h_q^p & h_q^{p+1} & \ldots & h_q^{p+q-2} \end{vmatrix} + O\left(h^{p+q-1}\right).$$

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 $x_i = x_0 + ih_i$ $(i = 0, 1, \ldots, I)$, $y_j = y_0 + jh_2$ $(j = 0, 1, \ldots, J)$. В табл. 3 представлена часть данных, которые нам в дальнейшем понадобятся.

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

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

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

Воспользовавшись введенными выше обозначениями, получим следующие приближенные выражения (аппрокси-

мации) для частных производных в узле (x_i, y_j) с помощью отношений конечных разностей:

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

Для численного дифференцирования функций многих переменных можно, как и ранее, использовать интерполяционные многочлены. Однако рассмотрим здесь другой

Таблица 3

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}	$u_{i-2,\,j+2}$	$u_{i-1,\,j+2}$	$u_{i,\,j+2}$	$u_{i+1,\,j+2}$	$u_{i+2,\,j+2}$

способ — разложение в ряд Тейлора функции двух переменных:

$$f(x + \Delta x,\ y + \Delta y) = f(x,\ y) + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y + \\ + \frac{1}{2!}\left(\frac{\partial^2 f}{\partial x^2}\Delta x^2 + 2\frac{\partial^2 f}{\partial x \partial y}\Delta x \Delta y + \frac{\partial^2 f}{\partial y^2}\Delta y^2\right) + \\ + \frac{1}{3!}\left(\frac{\partial^3 f}{\partial x^3}\Delta x^3 + 3\frac{\partial^3 f}{\partial x^2 \partial y}\Delta x^2 \Delta y + \right. \\ \left. + 3\frac{\partial^3 f}{\partial x \partial y^2}\Delta x \Delta y^2 + \frac{\partial^3 f}{\partial y^3}\Delta y^3\right) + \ldots \quad (3.22)$$

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:

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

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 (3.22) 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 комбинация используемых узлов (*шаблон*), которые отмечены кружочками. Значения производных вычисляются в узле (x_i, y_j), отмеченном крестиком (напомним, что на шаблонах и в табл. 3 по горизонтали изменяются переменная x и индекс i, по вертикали — переменная y и индекс j):

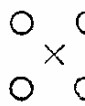
$$\text{○×○}\quad \left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h_1},$$

$$\begin{matrix}\text{○}\\ \text{×}\\ \text{○}\end{matrix}\quad \left(\frac{\partial u}{\partial y}\right)_{ij} = \frac{u_{i,j+1} - u_{i,j-1}}{2h_2},$$

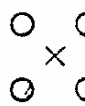
$$\text{○⊗○}\quad \left(\frac{\partial^2 u}{\partial x^2}\right)_{ij} = \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_1^2},$$

$$\begin{matrix}\text{○}\\ \text{⊗}\\ \text{○}\end{matrix}\quad \left(\frac{\partial^2 u}{\partial y^2}\right)_{ij} = \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h_2^2},$$

$$\begin{matrix}\text{○}\ \ \text{○}\\ \ \ \text{×}\\ \text{○}\ \ \text{○}\end{matrix}\quad \left(\frac{\partial^2 u}{\partial x \partial y}\right)_{ij} = \frac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{4h_1 h_2}.$$



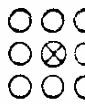
$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{u_{i+1,j+1} - u_{i-1,j+1} + u_{i+1,j-1} - u_{i-1,j-1}}{4h_1},$$



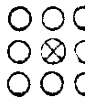
$$\left(\frac{\partial u}{\partial y}\right)_{ij} = \frac{u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j+1} - u_{i-1,j-1}}{4h_2},$$

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

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



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



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

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 $\Delta x_i = x_i - x_{i-1}$:

$$s_i = f(\xi_i)\,\Delta x_i. \qquad (3.23)$$

Составим сумму всех таких произведений:

$$S_n = s_1 + s_2 + \ldots + s_n = \sum_{i=1}^{n} f(\xi_i)\,\Delta x_i. \qquad (3.24)$$

Сумма S_n называется *интегральной суммой. Определенным интегралом* от функции $f(x)$ на отрезке $[a, b]$ называется предел интегральной суммы при неограниченном увеличении числа точек разбиения; при этом длина наибольшего из элементарных отрезков стремится к нулю:

$$\int_a^b f(x)\,dx = \lim_{\max \Delta x_i \to 0} \sum_{i=1}^{n} f(\xi_i)\,\Delta x_i. \qquad (3.25)$$

Теорема 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.*

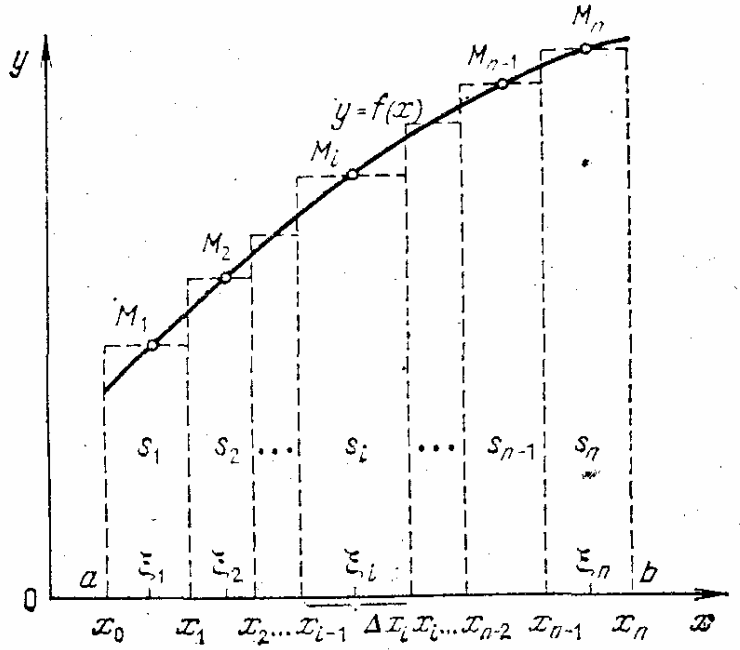


Рис. 11

Геометрический смысл введенных понятий для случая $f(x) > 0$ проиллюстрирован на рис. 11. Абсциссами точек M_i являются значения ξ_i, ординатами — значения $f(\xi_i)$. Выражения (3.23) при $i = 1, 2, \ldots, n$ описывают площади элементарных прямоугольников (штриховые линии), интегральная сумма (3.24) — 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 (3.25).

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:

$$\int_a^b f(x)\,dx = F(x)\Big|_a^b = F(b) - F(a). \qquad (3.26)$$

Однако на практике этой формулой часто нельзя воспользоваться по двум основным причинам: 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 методы численного 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.

Пример. Вычислить интеграл $I = \int_0^1 e^{-x^2}\,dx$ с погрешностью 10^{-4}.

Воспользуемся разложением экспоненты в ряд:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

Используя последнее выражение и заменяя x на $-x^2$,

записываем интеграл в виде

$$I = \int_0^1 \left(1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \ldots\right) dx =$$

$$= x - \frac{x^3}{3} + \frac{x^5}{5 \cdot 2!} - \frac{x^7}{7 \cdot 3!} + \ldots \Big|_0^1 =$$

$$= 1 - \frac{1}{3} + \frac{1}{10} - \frac{1}{42} + \ldots \approx 0.7468.$$

Более универсальными методами, которые пригодны для обоих случаев, являются *методы численного интегрирования*, основанные на аппроксимации подынтегральной функции с помощью интерполяционных многочленов. В дальнейшем будем использовать кусочную (локальную) интерполяцию. Это позволит приближенно заменить определенный интеграл интегральной суммой (3.24). В зависимости от способа ее вычисления получаются разные методы численного интегрирования (методы прямоугольников, трапеций, парабол, сплайнов и др.).

Следует отметить, что к вычислению определенного интеграла сводятся многие практические задачи: вычисление площади фигур, определение работы переменной силы и др. Решение задач с использованием кратных интегралов также 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 (3.24). 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:

$$\int_a^b f(x)\,dx = h_1 y_0 + h_2 y_1 + \ldots + h_n y_{n-1}, \qquad (3.27)$$

$$\int_a^b f(x)\,dx = h_1 y_1 + h_2 y_2 + \ldots + h_n y_n. \qquad (3.28)$$

Широко распространенным и более точным является вид формулы прямоугольников, использующий значения

функции в средних точках элементарных отрезков (в *полуцелых узлах*):

$$\int_a^b f(x)\,dx = \sum_{i=1}^{n} h_i f(x_{i-1/2}), \tag{3.29}$$

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

В дальнейшем под методом прямоугольников будем понимать последний алгоритм (он еще называется *методом средних*).

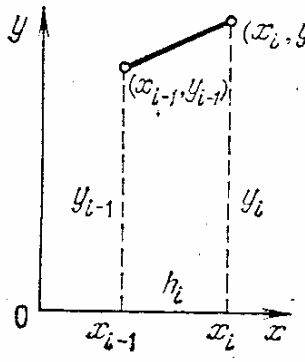


Рис. 12

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 (криволинейной трапеции) складывается из площадей элементарных прямолинейных трапеций (рис. 12). Площадь каждой такой трапеции равна произведению полусуммы оснований на высоту:

$$s_i = \frac{y_{i-1} + y_i}{2} h_i, \quad i = 1, 2, \ldots, n.$$

Складывая все эти равенства, получаем формулу трапеций для численного интегрирования:

$$\int_a^b f(x)\,dx = \frac{1}{2} \sum_{i=1}^{n} h_i (y_{i-1} + y_i). \tag{3.30}$$

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

$$\int_a^b f(x)\,dx = h \sum_{i=1}^{n} f(x_{i-1/2}), \tag{3.31}$$

$$\int_a^b f(x)\,dx = h\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right). \tag{3.32}$$

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 (3.26):

$$I = \operatorname{arctg} x \Big|_0^1 = \frac{\pi}{4} \approx 0.785398.$$

Используем теперь для вычисления данного интеграла формулы прямоугольников и трапеций. Разобьем отрезок интегрирования [0, 1] на десять равных частей: $n = 10$,

Таблица 4

x_i	y_i	$x_{i-1/2}$	$y_{i-1/2}$
0.0	1.000000		
0.1	0.990099	0.05	0.997506
0.2	0.961538	0.15	0.977995
0.3	0.917431	0.25	0.941176
0.4	0.862069	0.35	0.890868
0.5	0.800000	0.45	0.831601
0.6	0.735294	0.55	0.767754
0.7	0.671141	0.65	0.702988
0.8	0.609756	0.75	0.640000
0.9	0.552486	0.85	0.580552
1.0	0.500000	0.95	0.525624

$h = 0.1$. 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. 4). По формуле прямоугольников (3.31) получим

$$I_1 = h \sum_{i=1}^{10} y_{i-1/2} =$$

$$= 0.1\,(0{,}997506 + \ldots + 0{,}525624) = 0{,}785606,$$

Погрешность в вычислении интеграла составляет $\Delta I_1 = I_1 - I = 0.00021$ (около 0.027%). Используя формулу трапеций (3.32), находим $I_2 = 0.1(0.750000 + 0.990099 + \ldots + 0.552486) = 0.784981$. 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 s_i, использующим значения функции в центральной точке $x_{i-1/2}$ на отрезке $[x_{i-1}, x_i]$. Заметим, что использование формул прямоугольников в виде (3.27) или (3.28) приведет к погрешности более 3%.

В рассмотренном примере погрешность численного интегрирования легко оценивалась, поскольку имелось точное значение интеграла. В общем случае погрешность R_n численного значения S_n равна

$$R_n = \int_a^b f(x)\,dx - S_n.$$

Она зависит от шага разбиения, и ее можно представить в виде $R_n = O(h^k)$. В случае переменного шага можно принять $h = \max h_i$. Из этого представления погрешности численного интегрирования следует, что при $h \to 0$ $(n \to \infty)$ значения интеграла, получаемые путем численного интегрирования, сходятся к его точному значению. Заметим, что это имеет место, если подынтегральная функция на конечном отрезке $[a, b]$ интегрируема (достаточное условие).

На основании формул прямоугольников и трапеций можно получить *уточненные значения интегралов*, если учесть характер погрешностей этих формул. Главный член погрешности формулы прямоугольников (3.29) на каждом отрезке $[x_{i-1}, x_i]$ равен $\frac{1}{24} h_i^3 f^4(x_{i-1/2})$; для формулы трапеций он равен $-\frac{1}{12} h_i^3 f''(x_i)$, т. е. примерно вдвое больше и имеет другой знак. На основании этого можно записать уточненную формулу для вычисления определенного интеграла с использованием значений I_1

и I_2, вычисленных по методам прямоугольников и трапеций:

$$I \approx (2I_1 + I_2)/3. \qquad (3.33)$$

Для рассмотренного выше примера получено $I_1 = 0.785606$, $I_2 = 0.784981$. Поэтому, следуя (3.33), найдем $I = (2 \cdot 0.785606 + 0.784981)/3 = 0.785398$ (с точностью до погрешностей округления), т. е. все шесть разрядов равны точным значениям.

Поскольку погрешность численного интегрирования определяется шагом разбиения, то, уменьшая его, 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. Розобьем 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:

$$f(x) \approx \varphi_i(x) = a_i x^2 + b_i x + c_i,$$

$$x_{i-1} \leqslant x \leqslant x_{i+1}.$$

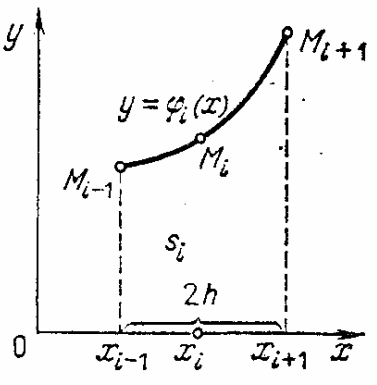


Рис. 13

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 $M_{i-1}(x_{i-1}, y_{i-1})$, $M_i(x_i, y_i)$, $M_{i+1}(x_{i+1}, y_{i+1})$:

$$\varphi_i(x) = \frac{(x - x_i)(x - x_{i+1})}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})} y_{i-1} +$$

$$+ \frac{(x - x_{i-1})(x - x_{i+1})}{(x_i - x_{i-1})(x_i - x_{i+1})} y_i + \frac{(x - x_{i-1})(x - x_i)}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} y_{i+1}.$$

Элементарная площадь s_i (рис. 13) может быть вычислена с помощью определенного интеграла. Учитывая

равенства $x_{i+1} - x_i = x_i - x_{i-1} = h$, получаем

$$s_i = \int_{x_{i-1}}^{x_{i+1}} \varphi_i(x)\,dx = \frac{1}{2h^2}\int_{x_{i-1}}^{x_{i+1}} [(x - x_i)(x - x_{i+1})\,y_{i-1} -$$

$$- 2(x - x_{i-1})(x - x_{i+1})\,y_i + (x - x_{i-1})(x - x_i)\,y_{i+1}]\,dx =$$

$$= \frac{h}{3}(y_{i-1} + 4y_i + y_{i+1}).$$

Проведя такие вычисления для каждого элементарного отреза $[x_{i-1}, x_{i+1}]$, просуммируем полученные выражения:

$$S = \frac{h}{3}(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots$$

$$\ldots + 2y_{n-2} + 4y_{n-1} + y_n).$$

Данное выражение для S принимается в качестве значения определенного интеграла:

$$\int_a^b f(x)\,dx \approx \frac{h}{3}[y_0 + 4(y_1 + y_3 + \ldots + y_{n-1}) +$$

$$+ 2(y_2 + y_4 + \ldots + y_{n-2}) + y_n]. \quad (3.34)$$

Полученное соотношение называется *формулой Симпсона.*

Эту формулу можно получить и другими способами, например комбинированием формул прямоугольников и трапеций или двукратным применением метода трапеций при разбиениях отрезка $[a, b]$ на части с шагами h и $2h$. При этом важно добиться, чтобы главные части погрешностей этих методов при сложении уничтожались. В частности, комбинация формул прямоугольников и трапеций, определяемая соотношением (3.33), аналогична формуле Симпсона.

Пример. 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. 4.

Применяя формулу (3.34), 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, как и по формуле (3.33), оказался совпадающим с точным значением (шесть значащих цифр).

Сравнив методы прямоугольников и трапеций с методом Симпсона, отметим, что последний обладает более высокой точностью. Главный член погрешности метода Симпсона имеет вид

$$R_n = -\frac{h^4}{180} f^{IV}(x).$$

Напомним, что погрешность методов прямоугольников и трапеций имеет порядок $O(h^3)$, а уточненная формула (3.33) построена так, что коэффициент при h^3 в выражении для погрешности обращается в нуль. Таким образом, погрешности метода Симпсона и формулы (3.33), использующей методы прямоугольников и трапеций, имеют один порядок.

Отметим разницу между формулой Симпсона (3.34) и комбинацией методов прямоугольников и трапеций (3.33). Эти формулы имеют одинаковую погрешность, однако формула (3.33) требует двукратного вычисления интеграла разными методами. Кроме того, для метода Симпсона нужно почти вдвое меньше табличных значений функции, поскольку для метода прямоугольников нужны дополнительные данные в полуцелых точках.

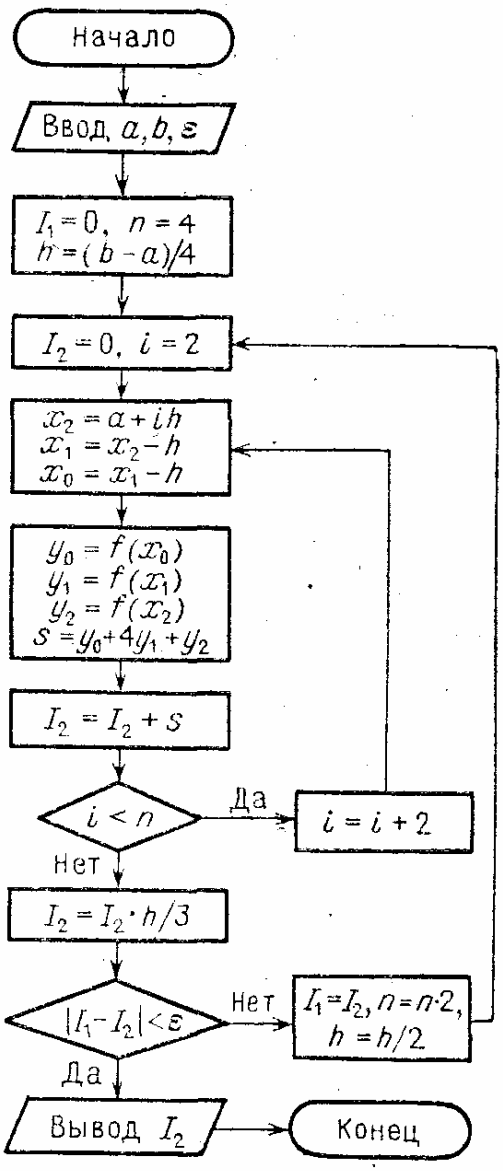


Рис. 14. Блок-схема метода Симпсона

Блок-схема одного из простейших алгоритмов вычисления определенного интеграла по методу Симпсона представлена на рис. 14. В качестве исходных данных задаются границы отрезка интегрирования a, b, погреш-

ность ε, а также формула для вычисления значений подынтегральной функции $y=f(x)$. Первоначально отрезок $[a, b]$ разбивается на четыре части с шагом $h=(b-a)/4$. Вычисляется значение интеграла I_1. Потом число шагов удваивается, вычисляется значение I_2 с шагом $h/2$. Условие окончания 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. 14 алгоритм не является оптимальным. В частности, при вычислении каждого последующего приближения I_2 не используются значения функции $f(x)$, уже найденные на предыдущем этапе. Более экономичные алгоритмы будут рассмотрены в п. 5.

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. 2).

Разобьем отрезок интегрирования $[a, b]$ на n частей точками x_i. Пусть $x_i-x_{i-1}=h_i$ $(i=1, 2, \ldots, n)$. На каждом элементарном отрезке интерполируем подынтегральную функцию $f(x)$ с помощью кубического сплайна:

$$\varphi_i(x)=a_i+b_i(x-x_{i-1})+c_i(x-x_{i-1})^2+d_i(x-x_{i-1})^3, \tag{3.35}$$

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

Выражение для интеграла представим в виде

$$I=\int_a^b f(x)\,dx=\sum_{i=1}^{n}\int_{x_{i-1}}^{x_{i+1}} f(x)\,dx\approx\sum_{i=1}^{n}\int_{x_{i-1}}^{x_{i+1}}\varphi_i(x)\,dx.$$

Используя выражение (3.35), в результате вычисления интегралов находим

$$I\approx\sum_{i=1}^{n}\left(a_ih_i+\frac{1}{2}b_ih_i^2+\frac{1}{3}c_ih_i^3+\frac{1}{4}d_ih_i^4\right). \tag{3.36}$$

Способ вычисления коэффициентов a_i, b_i, c_i, d_i описан в гл. 2. Здесь лишь отметим, что $a_i=y_{i-1}$.

Для практических расчетов формулу (3.36) можно представить в виде

$$I\approx\frac{1}{2}\sum_{i=1}^{n}h_i(y_{i-1}+y_i)-\frac{1}{12}\sum_{i=1}^{n}h_i^3(c_{i-1}+c_i). \tag{3.37}$$

Анализ этой формулы показывает, что первый член в правой части совпадает с правой частью формулы (3.30) для метода трапеций. Следовательно, второй член характеризует поправку к методу трапеций, которую дает использование сплайнов.

Как следует из формулы (3.35), 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 (3.37):

$$\frac{h_i^3}{12}(c_{i-1} + c_i) \approx \frac{h_i^3}{12}y''_*,$$

где y''_* — вторая производная в некоторой внутренней точке. Полученная оценка показывает, что добавка к формуле трапеций, которую дает использование сплайнов, компенсирует погрешность самой формулы трапеций.

Отметим, что во всех предыдущих методах (см. п. 2, 3) формулы численного интегрирования можно условно записать в виде линейной комбинации табличных значений функции:

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} \alpha_i y_i.$$

При использовании сплайнов такое представление невозможно, поскольку сами коэффициенты α_i зависят от всех значений y_i.

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. 14 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 метода $O(h^p)$ 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

$$I = I_1 - \frac{(I_1 - I_2)^2}{I_1 - 2I_2 + I_3},$$

а порядок точности используемого метода численного интегрирования определяется соотношением

$$p = \frac{1}{\ln q} \ln \frac{I_3 - I_2}{I_2 - I_1}.$$

Уточнение значения интеграла можно также проводить методом Рунге — Ромберга (см. § 1, п. 5).

5. 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 функция меняется 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 стандарт-

ной подпрограммы в математическое обеспечение ЭВМ. 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.

$$\left| I - \int_a^b f(x)\,dx \right| \leqslant \varepsilon. \tag{3.38}$$

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 число шагов, их расположение и размеры зависят 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 $I_i^{(1)}$, $I_i^{(2)}$ для интеграла по этому отрезку:

$$I_i = \int_{x_{i-1}}^{x_i} f(x)\,dx. \tag{3.39}$$

Полученные значения сравниваем и проводим оценку их погрешности. Если погрешность находится в допустимых границах, то одно из этих приближений принимается за значение интеграла по этому 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 (3.39) 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 $h_i/4$. Получим выражения

$$I_i^{(1)} = \frac{h_i}{6}\left[f(x_{i-1}) + 4f\left(x_{i-1} + \frac{h_i}{2}\right) + f(x_i)\right], \tag{3.40}$$

$$I_i^{(2)} = \frac{h_i}{12}\left[f(x_{i-1}) + 4f\left(x_{i-1} + \frac{h_i}{4}\right) + \right.$$
$$\left. + 2f\left(x_{i-1} + \frac{h_i}{2}\right) + 4f\left(x_i + \frac{3h_i}{4}\right) + f(x_i)\right]. \tag{3.41}$$

Формулу (3.41) можно также получить двукратным применением формулы (3.40) для отрезков $[x_{i-1},\ x_{i-1} + h_i/2]$ и $[x_{i-1} + h_i/2,\ x_i]$.

Процесс деления отрезка пополам и вычисления уточненных значений $I_i^{(1)}$ и $I_i^{(2)}$ продолжается до тех пор, пока их разность станет не больше некоторой заданной величины δ_i, зависящей от ε и h:

$$|I_i^{(1)} - I_i^{(2)}| \leqslant \delta_i. \tag{3.42}$$

Аналогичная процедура проводится для всех n элементарных отрезков. Величина $I = \sum_{i=1}^{n} I_i$ принимается в качестве искомого значения интеграла. Условия (3.42) и соответствующий выбор величин δ_i обеспечивают выполнение условия (3.38).

6. 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, зависящей от числа узлов. Точность формул растет с увеличением степени интерполяционного многочлена *).

Метод Гаусса не предполагает разбиения отрезка интегрирования на равные промежутки. Формулы численного интегрирования интерполяционного типа ищутся та-

*) Заметим, что формулы прямоугольников, трапеций и Симпсона являются частными случаями формул Ньютона — Котеса.

кими, 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

$$\int_{-1}^{1} \frac{f(x)\,dx}{\sqrt{1-x^2}}.$$

Получающаяся формула характерна тем, что все коэффициенты при y_i 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

$$\int_{-1}^{1} p(x) f(x)\,dx = k \sum_{i=0}^{n} y_i + R.$$

При этом решается следующая задача: найти точки x_0, x_1, ..., x_n и коэффициент k такие, при которых остаточный член R обращается в нуль, когда функция $f(x)$ является произвольным многочленом возможно большей степени.

Формула Эйлера использует не только значения подынтегральной функции в точках разбиения, но и ее производные до некоторого порядка на границах отрезка.

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 ($k = 1, 2, \ldots$) 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 $x = c$ ($a \leqslant c \leqslant b$) 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 интервал интегрирования $[a, \infty)$ 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 b так, чтобы принятое значение интеграла отличалось от исходного на некоторый малый остаток, т. е.

$$\int_a^{\infty} f(x)\,dx = \int_a^{b} f(x)\,dx + R, \quad R = \int_b^{\infty} f(x)\,dx.$$

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 непосредственно по формулам. Тогда численный метод используется только для интегрирования ограниченной функции $\varphi(x)$.

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).

7. Кратные интегралы. Численные методы используются также для вычисления кратных интегралов. Ограничимся здесь рассмотрением *двойных интегралов* вида

$$I = \int\int_G f(x, y)\, dx\, dy. \tag{3.43}$$

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 $f(x, y)$:

$$\bar{f}(x, y) = \frac{1}{S}\int\int_G f(x, y)\, dx\, dy, \quad S = (b - a)(d - c). \tag{3.44}$$

Будем считать, что среднее значение приближенно равно значению функции в центре прямоугольника, т. е.

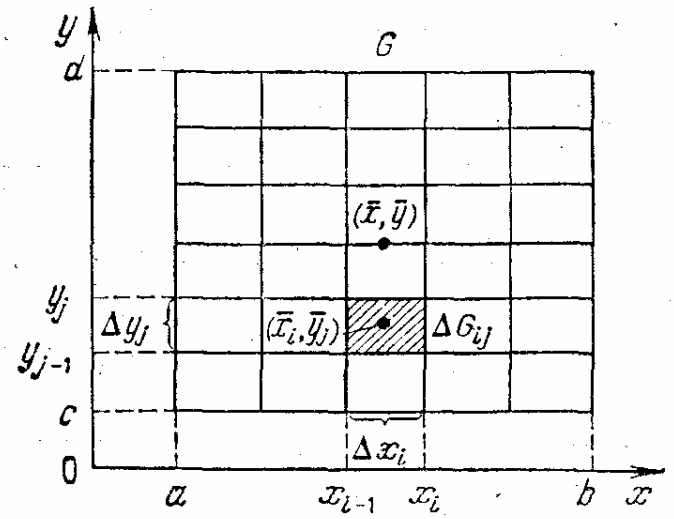


Рис. 15.

$\bar{f}(x, y) = f(\bar{x}, \bar{y})$. Тогда из (3.44) получим выражение для приближенного вычисления двойного интеграла:

$$\int\int_G f(x, y)\, dx\, dy \approx S f(\bar{x}, \bar{y}),$$
$$\bar{x} = (a + b)/2, \quad \bar{y} = (c + d)/2. \tag{3.45}$$

Точность этой формулы можно повысить, если разбить область G на прямоугольные ячейки ΔG_{ij} (рис. 15): $x_{i-1} \leqslant x \leqslant x_i$ $(i = 1, 2, \ldots, M)$, $y_{j-1} \leqslant y \leqslant y_j$ $(j = 1, 2, \ldots, N)$. Применяя к каждой ячейке формулу (3.45),

получаем

$$\iint\limits_{\Delta G_{ij}} f(x, y)\,dxdy \approx f(\bar{x}_i, \bar{y}_j)\,\Delta x_i \Delta y_j.$$

Суммируя эти выражения по всем ячейкам, находим значение двойного интеграла:

$$\iint\limits_{G} f(x, y) = \sum_{i=1}^{M} \sum_{j=1}^{N} f(\bar{x}_i, \bar{y}_j)\,\Delta x_i \Delta y_j. \qquad (3.46)$$

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

$$R \approx O(1/M^2 + 1/N^2) \approx O(\Delta x^2 + \Delta y^2).$$

Таким образом, формула (3.46) имеет второй порядок точности. Для повышения точности 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 (3.46) 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 (3.43) для прямоугольной области можно записать в виде

$$\iint\limits_G f(x, y)\,dx\,dy = \int\limits_a^b F(x)\,dx, \qquad F(x) = \int\limits_c^d f(x, y)\,dy.$$

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.

8. 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,* 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 $\xi_1, \xi_2, \ldots, \xi_n$, и по совокупности этих значений приближенно определяется искомая величина

$$\xi = (\xi_1 + \xi_2 + \ldots + \xi_n)/n \approx x,$$

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

Пусть η — равномерно распределенная на 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

$$M\xi = \int_{-\infty}^{\infty} f(x) P_\eta(x)\, dx = \int_0^1 f(x)\, dx.$$

Следовательно, читая это равенство в обратном порядке, приходим к выводу, что интеграл $\int_0^1 f(x)\, dx$ может быть вычислен как математическое ожидание некоторой случайной величины ξ, которая определяется независимыми реализациями η_i случайной величины η с равномерным законом распределения:

$$\int_0^1 f(x)\, dx \approx \xi = \frac{1}{n} \sum_{i=1}^{n} f(\eta_i).$$

Аналогично могут быть вычислены и кратные интегралы. Для двойного интеграла получим

$$\int\int_G f(x, y)\, dxdy \approx \frac{1}{n} \sum_{i=1}^{n} f(\eta_i, \zeta_i),$$

где G: $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$; η_i, ζ_i — независимые реализации случайных величин η, ζ, 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 ЭВМ этот способ непригоден, поскольку

трудно дважды получить одинаковые совокупности случайных чисел, которые необходимы при отладке программ.

Известны многие таблицы случайных чисел, которые вычислялись независимо. Их можно вводить в ЭВМ, хранить в виде файла на магнитной ленте или магнитном диске коллективного пользования. А еще лучше заготовить собственный файл случайных чисел.

В настоящее время наиболее распространенный способ выработки случайных чисел на ЭВМ состоит в том, что в памяти хранится некоторый алгоритм выработки 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.

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 и третьим порядками точности.

2. Составить блок-схему алгоритма вычисления производной функции, заданной таблицей с постоянным шагом на некотором отрезке.

3. Вычислить $\int\limits_0^1 e^{x^2} dx$, используя методы прямоугольников, трапеций и Симпсона. Отрезок интегрирования разделить на десять равных частей.

4. 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.

5. Составить блок-схему решения упр. 4.

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*:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n. \end{aligned} \tag{4.1}$$

Совокупность коэффициентов этой системы запишем в виде таблицы:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}. \tag{4.2}$$

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 матричном виде:

$$AX = B, \tag{4.3}$$

где X и B — вектор-столбец неизвестных и вектор-столбец правых частей соответственно:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{bmatrix}.$$

В ряде случаев получаются системы уравнений с некоторыми специальными видами матриц. Вот некоторые примеры таких матриц:

$$A = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 3 & 2 \\ -1 & 2 & 4 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \\ 0 & 0 & 2 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 \\ 2 & -1 & 2 & 0 & 0 & 0 \\ 3 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & -1 & 1 \\ 0 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 2 & 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 3 & 2 & 0 & 0 & 0 & 0 \\ 1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 3 & -2 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 & 3 & 1 \\ 0 & 0 & 0 & 0 & -1 & 3 \end{bmatrix},$$

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad O = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

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)); D — ленточная матрица (ее ненулевые элементы составляют «ленту», параллельную диагонали (в данном случае ленточная матрица D одновременно является также трехдиагональной)); E — единичная матрица (частный случай диагональной); O — нулевая матрица.

Определителем (*детерминантом*) матрицы A n-го порядка называется число D ($\det A$), равное

$$D = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = \sum (-1)^k a_{1\alpha} a_{2\beta} \dots a_{n\omega}. \quad (4.4)$$

Здесь индексы α, β, ..., ω пробегают все возможные $n!$ перестановок номеров 1, 2, ..., n; k — число инверсий в данной перестановке.

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 бесчисленное множество.

Все эти случаи легко проиллюстрировать геометрически для системы

$$\begin{aligned} a_1x + b_1y &= c_1, \\ a_2x + b_2y &= c_2. \end{aligned} \tag{4.5}$$

Каждое уравнение описывает прямую на плоскости; координаты точки пересечения указанных прямых являются решением системы (4.5).

Рассмотрим три возможных случая взаимного расположения двух прямых на плоскости:

1) прямые пересекаются — коэффициенты системы (4.5) не пропорциональны:

$$\frac{a_1}{a_2} \neq \frac{b_1}{b_2}; \tag{4.6}$$

2) прямые параллельны — коэффициенты системы (4.5) подчиняются условиям

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}; \tag{4.7}$$

3) прямые совпадают — все коэффициенты (4.5) пропорциональны:

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}. \tag{4.8}$$

Запишем определитель D системы (4.5) в виде

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}.$$

Отметим, что при выполнении условия (4.6) $D \neq 0$, и система (4.5) имеет единственное решение. В случаях отсутствия решения или при бесчисленном множестве решений имеют место соответственно соотношения (4.7) или (4.8), из которых получаем $D = 0$.

На практике, особенно при вычислениях на ЭВМ, 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.

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 диагональной матрицей с элементами $a_{ii} = 0.1$ не является плохо обусловленной, хотя ее определитель мал ($D = 10^{-n}$).

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 ЭВМ 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. 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 сравнительно небольших ($n < 200$) 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 выполнении вычислений с бесконечным числом разрядов (разумеется, при точных значениях коэффициентов системы). На практике при использовании ЭВМ вычисления проводятся с ограниченным числом знаков, определяемым разрядностью машины. Поэтому неизбежны погрешности 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

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12},$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{21}a_{32}a_{13} - a_{31}a_{22}a_{13} - a_{21}a_{12}a_{33} - a_{32}a_{23}a_{11}.$$

Определитель треугольной матрицы равен произведению ее элементов, расположенных на главной диагонали: $D = a_{11}a_{22}\ldots a_{nn}$. 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 (4.4)

$$D = \sum (-1)^h a_{1\alpha}a_{2\beta}\ldots a_{n\omega}.$$

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

$$N = n \cdot n! - 1 \approx n \cdot n!. \qquad (4.9)$$

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 ЭВМ с заданным быстродействием. Примем для определенности среднее быстродействие равным 100 000 операций в секунду. Тогда для вычисления определителя 10-го порядка потребуется около 6 мин, а при $n = 20$ — около $1.4 \cdot 10^{11}$ ч, т. е. свыше $5 \cdot 10^9$ сут. 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

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i1} & \cdots & a_{ij} & \cdots & a_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nj} & \cdots & a_{nn} \end{bmatrix}.$$

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.

$$A_{ij} = (-1)^{i+j} \begin{vmatrix} a_{11} & \cdots & a_{1,j-1} & a_{1,j+1} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{i-1,1} & \cdots & a_{i-1,j-1} & a_{i-1,j+1} & \cdots & a_{i-1,n} \\ a_{i+1,1} & \cdots & a_{i+1,j-1} & a_{i+1,j+1} & \cdots & a_{i+1,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{n,j-1} & a_{n,j+1} & \cdots & a_{nn} \end{vmatrix}.$$

Каждый элемент b_{ij} $(i, j = 1, \ldots, n)$ обратной матрицы $B = A^{-1}$ равен отношению алгебраического дополнения A_{ji} элемента a_{ji} (не a_{ij}) исходной матрицы A к значению ее определителя D:

$$B = A^{-1} = \begin{bmatrix} \frac{A_{11}}{D} & \frac{A_{21}}{D} & \cdots & \frac{A_{n1}}{D} \\ \frac{A_{12}}{D} & \frac{A_{22}}{D} & \cdots & \frac{A_{n2}}{D} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{A_{1n}}{D} & \frac{A_{2n}}{D} & \cdots & \frac{A_{nn}}{D} \end{bmatrix}. \qquad (4.10)$$

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

$$N=[(n-1)\cdot(n-1)!-1]n^2+n^2+n\cdot n!-1=n^2\cdot n!-1. \tag{4.11}$$

Важной задачей линейной алгебры является также вычисление собственных значений матрицы. Этому вопросу будет посвящен отдельный параграф.

§ 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

$$a_1x+b_1y=c_1,$$
$$a_2x+b_2y=c_2.$$

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 (4.9). 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 $AX = B$ (см. (4.3)). Тогда, умножая обе части этого векторного уравнения слева на обратную матрицу A^{-1}, получаем $X = A^{-1}B$. 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. (Заметим, что к такому виду приводится лишь невырожденная матрица. В противном случае метод Гаусса неприменим.)

Обратный ход метода Гаусса состоит в последовательном вычислении искомых неизвестных: решая последнее уравнение, находим единственное неизвестное x_n. Далее, используя это значение, из предыдущего уравнения вычисляем x_{n-1} и т. д. Последним найдем x_1 из первого уравнения.

Рассмотрим применение метода Гаусса для системы

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \qquad (4.12)$$

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 третье-

му уравнению, также исключим из него x_1. Получим равносильную систему уравнений вида

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a'_{22}x_2 + a'_{23}x_3 &= b'_2, \\ a'_{32}x_2 + a'_{33}x_3 &= b'_3; \end{aligned} \tag{4.13}$$

$$a'_{ij} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}, \quad i, j = 2, 3,$$

$$b'_i = b_i - \frac{a_{i1}}{a_{11}} b_1, \qquad i = 2, 3.$$

Теперь из третьего уравнения системы (4.13) нужно исключить x_2. Для этого умножим второе уравнение на $-a'_{32}/a'_{22}$ и прибавим результат к третьему. Получим

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a'_{22}x_2 + a'_{23}x_3 &= b'_2, \\ a''_{33}x_3 &= b''_3; \end{aligned} \tag{4.14}$$

$$a''_{33} = a'_{33} - \frac{a'_{32}}{a'_{22}} a'_{23}, \quad b''_3 = b'_3 - \frac{a'_{32}}{a'_{22}} b'_2.$$

Матрица системы (4.14) 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 отличными от нуля; в противном случае необходимо соответственным образом переставить уравнения системы. Перестановка уравнений должна быть предусмотрена в вычислительном алгоритме при его реализации на ЭВМ.

Обратный ход начинается с решения третьего уравнения системы (4.14):

$$x_3 = b''_3/a''_{33}.$$

Используя это значение, можно найти x_2 из второго уравнения, а затем x_1 из первого:

$$x_2 = \frac{1}{a'_{22}} \left(b'_2 - a'_{23}x_3\right), \quad x_1 = \frac{1}{a_{11}} \left(b_1 - a_{12}x_2 - a_{13}x_3\right).$$

Аналогично строится вычислительный алгоритм для линейной системы с произвольным числом уравнений.

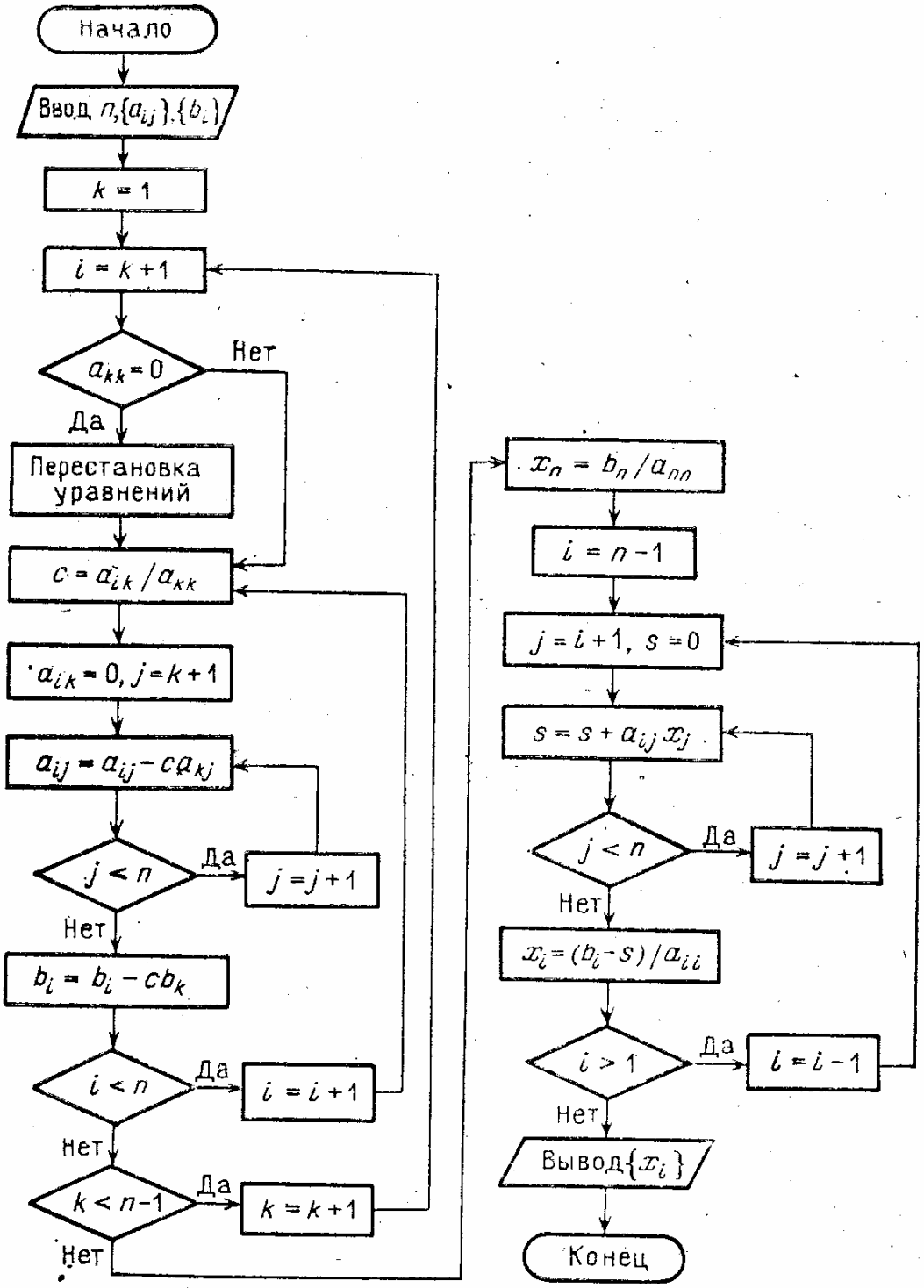


Рис. 16. Блок-схема метода Гаусса

На рис. 16 приведена блок-схема решения методом Гаусса системы n линейных уравнений вида

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1,\\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2,\\
\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ &\\
a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n.
\end{aligned}$$

Левая часть блок-схемы соответствует прямому ходу. Поясним смысл индексов: i — номер уравнения, из которого исключается неизвестное x_k; j — номер столбца; k — номер неизвестного, которое исключается из оставшихся $n-k$ уравнений (а также номер того уравнения, с помощью которого исключается x_k). Операция перестановки уравнений (т. е. перестановки соответствующих коэффициентов) служит для предотвращения деления на нулевой элемент. Правая часть блок-схемы описывает процесс обратного хода. Здесь i — номер неизвестного, которое определяется из i-го уравнения; $j = i+1,\ i+2,\ \ldots$ — номера уже найденных неизвестных.

Одной из модификаций метода Гаусса является *схема с выбором главного элемента*. Она состоит в том, что требование неравенства нулю диагональных элементов a_{kk}, на которые происходит деление в процессе исключения, заменяется более жестким: из всех оставшихся в k-м столбце элементов нужно выбрать наибольший по модулю и переставить уравнения так, чтобы этот элемент оказался на месте элемента a_{kk}.

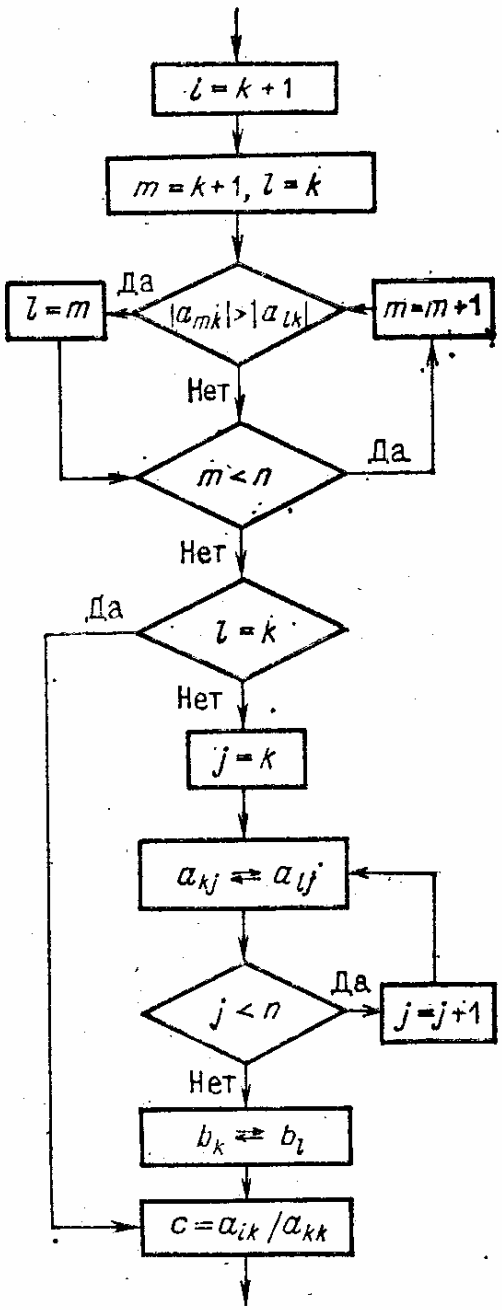


Рис. 17. Выбор главного элемента

Блок-схема алгоритма выбора главного элемента приведена на рис. 17. Она дополняет блок-схему метода Гаусса (см. рис. 16). Здесь введены новые индексы: l — номер наибольшего по абсолютной величине элемента матрицы в столбце с номером k (т. е. среди элементов $a_{kk}, \ldots, a_{km}, \ldots, a_{kn}$); m — текущий номер элемента, с которым происходит сравне-

ние. Заметим, что диагональные элементы матрицы называются *ведущими* элементами; ведущий элемент a_{kk} — это коэффициент при k-м неизвестном в k-м уравнении на k-м шаге исключения.

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 сравнительно небольшого числа ($n \leqslant \leqslant 100$) 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:

$$\begin{aligned} 10x_1 - 7x_2 &= 7, \\ -3x_1 + 2x_2 + 6x_3 &= 4, \\ 5x_1 - x_2 + 5x_3 &= 6. \end{aligned}$$

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 получаем число 15 002.5, которое нужно округлить до пяти разрядов. В результате

получаем третье уравнение в виде

$$15\,005x_3 = 15\,004.$$

Отсюда $x_3 = 15\,004/15\,005 = 0.99993$. Из второго и первого уравнений найдем

$$x_2 = \frac{6\cdot 0.99993 - 6.001}{0.001} - \frac{0.0015}{0.001} = -1.5,$$

$$x_1 = \frac{7(-1.5) + 7}{10} = -0.35.$$

Вычисления проводились с усечением до пяти разрядов по аналогии с процессом вычислений на ЭВМ. В результате этого было получено решение (—0.35, —1.5, 0.99993) вместо (0, —1, 1).

Такая большая неточность результатов объясняется малой величиной ведущего элемента. В подтверждение этому переставим сначала 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 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:

$$x_2 = \frac{2.5 - 5\cdot 1}{2.5} = -1, \quad x_1 = \frac{7 + 7(-1)}{10} = 0.$$

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: $AX = B$. Решение этой системы можно представить в виде $X = A^{-1}B$. Однако вычисленное по методу Гаусса решение 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* ε, равная разности этих значений: $\varepsilon = X - X_*$. Другая — *невязка* r, равная разности между правой и левой частями уравнений при подстановке в них решения: $r = B - AX_*$.

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 $\varepsilon \approx 0$ обычно $r \approx 0$, но обратное утверждение справедливо не всегда. В частности, для плохо обусловленных систем при $r \approx 0$ погрешность решения может быть большой.

Вместе с тем в практических расчетах, если система не является плохо обусловленной, контроль точности решения осуществляется с помощью невязки. Можно отметить, что метод Гаусса с выбором главного элемента в этих случаях дает малые невязки.

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

$$\det A = \pm \prod_{k=1}^{n} a_{kk}.$$

Здесь диагональные элементы a_{kk} берутся из преобразованной (а не исходной) матрицы. Знак зависит от того, четной или нечетной была суммарная перестановка 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 100-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

$$\sum_{k=1}^{n} a_{ik}z_{kj} = \delta_{ij}, \quad \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \quad i, j = 1, 2, \ldots, n. \tag{4.15}$$

Отсюда следует, что для нахождения элементов одного столбца обратной матрицы необходимо решить линейную систему (4.15) с матрицей A. Например, элементы j-го столбца $z_{1j}, z_{2j}, \ldots, z_{nj}$ могут быть найдены в результате решения системы уравнений

$$\begin{aligned} &a_{11}z_{1j} + a_{12}z_{2j} + \ldots + a_{1n}z_{nj} = 0, \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{j1}z_{1j} + a_{j2}z_{2j} + \ldots + a_{jn}z_{nj} = 1, \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{n1}z_{1j} + a_{n2}z_{2j} + \ldots + a_{nn}z_{nj} = 0. \end{aligned} \tag{4.16}$$

Следовательно, для обращения матрицы нужно n раз решить систему уравнений приведенного вида при $j = 1, 2, \ldots, n$. Поскольку матрица системы не меняется, то исключение неизвестных при использовании метода Гаусса (прямой ход) проводится только один раз. Для каждой системы делается только обратный ход после некоторых преобразований с использованием правых частей систем вида (4.16).

Оценки показывают, что это весьма экономичный способ обращения матрицы. Он требует примерно лишь в три раза больше действий, чем при решении одной системы уравнений.

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

$$\begin{aligned} b_1x_1 + c_1x_2 &= d_1, \\ a_2x_1 + b_2x_2 + c_2x_3 &= d_2, \\ a_3x_2 + b_3x_3 + c_3x_4 &= d_3, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n-1}x_{n-2} + b_{n-1}x_{n-1} + c_{n-1}x_n &= d_{n-1}, \\ a_nx_{n-1} + b_nx_n &= d_n. \end{aligned} \tag{4.17}$$

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.

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 том, что каждое неизвестное x_i выражается через x_{i+1} с помощью прогоночных коэффициентов A_i, B_i:

$$x_i = A_i x_{i+1} + B_i, \quad i = 1, 2, \ldots, n-1. \tag{4.18}$$

Из первого уравнения системы (4.17) найдем

$$x_1 = -\frac{c_1}{b_1} x_2 + \frac{d_1}{b_1}.$$

С другой стороны, по формуле (4.18) $x_1 = A_1 x_2 + B_1$. Приравнивая коэффициенты в обоих выражениях для x_1, получаем

$$A_1 = -c_1/b_1, \quad B_1 = d_1/b_1. \tag{4.19}$$

Из второго уравнения системы (4.17) выразим x_2 через x_3, заменяя x_1 по формуле (4.18):

$$a_2(A_1 x_2 + B_1) + b_2 x_2 + c_2 x_3 = d_2.$$

Отсюда найдем

$$x_2 = \frac{-c_2 x_3 + d_2 - a_2 B_1}{a_2 A_1 + b_2},$$

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 можно вычислить прогоночные коэффициенты для любого номера i:

$$A_i = -\frac{c_i}{e_i}, \quad B_i = \frac{d_i - a_i B_{i-1}}{e_i}, \tag{4.20}$$

$$e_i = a_i A_{i-1} + b_i, \quad i = 2, 3, \ldots, n-1.$$

Обратная прогонка состоит в последовательном вычислении неизвестных x_i. Сначала нужно найти x_n. Для этого воспользуемся выражением (4.18) при $i = n-1$ и

последним уравнением системы (4.17). 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 (4.18) и выражения для прогоночных коэффициентов (4.19), (4.20), последовательно вычисляем все неизвестные x_{n-1}, x_{n-2}, ..., x_1. Блок-схема решения системы линейных уравнений вида (4.17) приведена на рис. 18.

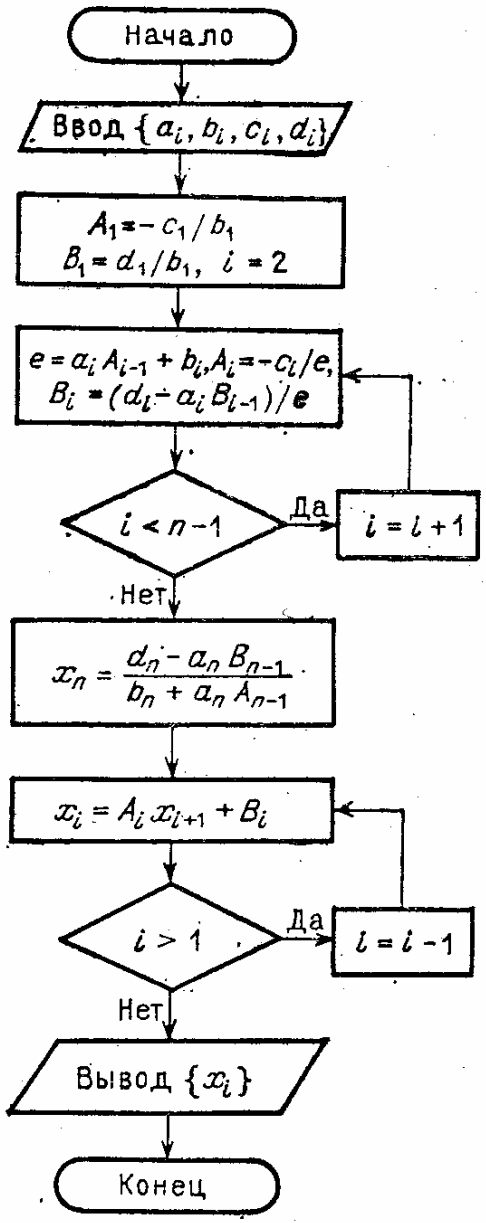


Рис. 18. Блок-схема метода прогонки

При анализе алгоритма метода прогонки надо учитывать возможность деления на нуль в формулах (4.20). 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 (4.17) 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 (4.17) метод прогонки оказывается устойчивым да-

же при нарушении условия преобладания диагональных элементов.

5. О других прямых методах. Среди прямых методов наиболее распространен метод Гаусса; он удобен для вычислений на ЭВМ. 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 ЭВМ 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

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2, \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ &, \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n. \end{aligned} \tag{4.21}$$

Пусть с помощью некоторого прямого метода вычислены приближенные значения неизвестных $x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$. Подставляя это решение в левые части системы (4.21), получаем некоторые значения $b_i^{(0)}$, отличные от b_i $(i = 1, 2, \ldots, n)$:

$$\begin{aligned} a_{11}x_1^{(0)} + a_{12}x_2^{(0)} + \ldots + a_{1n}x_n^{(0)} &= b_1^{(0)}, \\ a_{21}x_1^{(0)} + a_{22}x_2^{(0)} + \ldots + a_{2n}x_n^{(0)} &= b_2^{(0)}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n1}x_1^{(0)} + a_{n2}x_2^{(0)} + \ldots + a_{nn}x_n^{(0)} &= b_n^{(0)}. \end{aligned} \tag{4.22}$$

Введем обозначения: $\varepsilon_i^{(0)}$ — погрешности значений неизвестных, $r_i^{(0)}$ — невязки, т. е.

$$\varepsilon_i^{(0)} = x_i - x_i^{(0)}, \quad r_i^{(0)} = b_i - b_i^{(0)}, \quad i = 1, 2, \ldots, n. \tag{4.23}$$

Вычитая каждое уравнение системы (4.22) из соответствующего уравнения системы (4.21), с учетом обозначений (4.23) получаем

$$\begin{aligned} a_{11}\varepsilon_1^{(0)} + a_{12}\varepsilon_2^{(0)} + \ldots + a_{1n}\varepsilon_n^{(0)} &= r_1^{(0)}, \\ a_{21}\varepsilon_1^{(0)} + a_{22}\varepsilon_2^{(0)} + \ldots + a_{2n}\varepsilon_n^{(0)} &= r_2^{(0)}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n1}\varepsilon_1^{(0)} + a_{n2}\varepsilon_2^{(0)} + \ldots + a_{nn}\varepsilon_n^{(0)} &= r_n^{(0)}. \end{aligned} \tag{4.24}$$

Решая эту систему, находим значения погрешностей $\varepsilon_i^{(0)}$, которые используем в качестве поправок к решению. Следующие приближения неизвестных имеют вид

$$x_1^{(1)} = x_1^{(0)} + \varepsilon_1^{(0)}, \quad x_2^{(1)} = x_2^{(0)} + \varepsilon_2^{(0)}, \ \ldots, \ x_n^{(1)} = x_n^{(0)} + \varepsilon_n^{(0)}.$$

Таким же способом можно найти новые поправки к решению $\varepsilon_i^{(1)}$ и следующие приближения переменных $x_i^{(2)} = x_i^{(1)} + \varepsilon_i^{(1)}$ и т. д. Процесс продолжается до тех пор, пока все очередные значения погрешностей (поправок) ε_i не станут достаточно малыми.

Рассмотренный процесс уточнения решения представляет фактически итерационный метод решения системы линейных уравнений. При этом заметим, что для нахождения очередного приближения, т. е. на каждой итерации, решаются системы уравнений вида (4.24) с одной и той же матрицей, являющейся матрицей исходной системы (4.21), при разных правых частях. Это позволяет строить экономичесские алгоритмы. Например, при исполь-

зовании метода Гаусса сокращается объем вычислений на этапе прямого хода.

Решение систем уравнений с помощью рассмотренного метода, а также при использовании других итерационных методов сводится к следующему (рис. 19). Вводятся исходные данные, например коэффициенты уравнений и

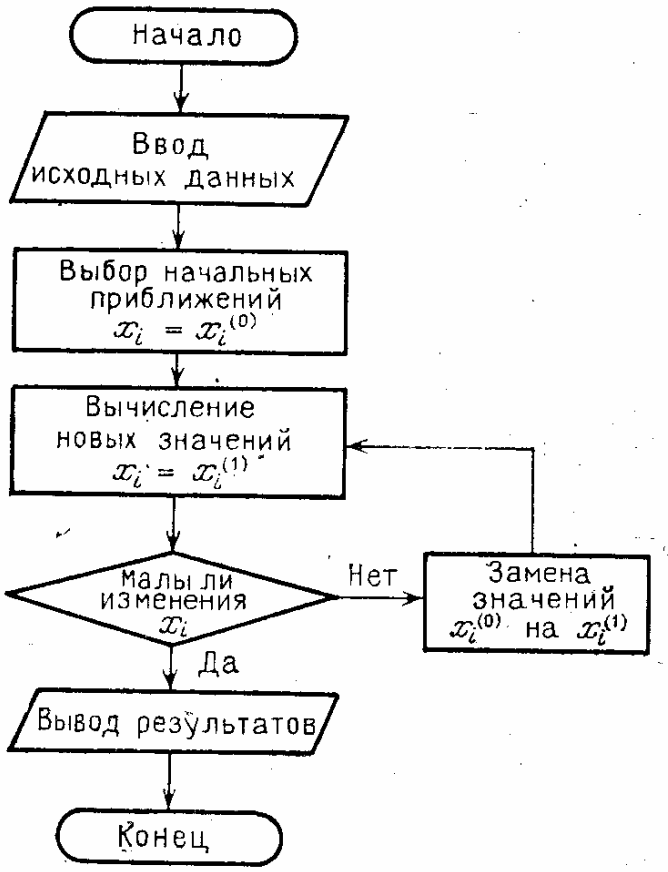


Рис. 19. Решение системы уравнений методом итераций

допустимое значение погрешности. Необходимо также задать начальные приближения значений неизвестных. Они либо вводятся в ЭВМ, 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. В результате каждой итерации получаются новые значения неизвестных. При малом (с заданной допустимой погрешностью) изменении этих значений на двух последовательных итерациях процесс прекращается, и происходит вывод значений неизвестных, полученных на последней итерации.

Заметим, что в этой схеме не предусмотрен случай отсутствия сходимости. Для предотвращения непроизво-

дительных 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 блок-схему.

2. Метод Гаусса — Зейделя. Одним из самых распространенных итерационных методов, отличающийся простотой и легкостью программирования, является *метод Гаусса — Зейделя.*

Проиллюстрируем сначала этот метод на примере решения системы

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3. \end{aligned} \tag{4.25}$$

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 (4.25):

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3), \tag{4.26}$$

$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3), \tag{4.27}$$

$$x_3 = \frac{1}{a_{33}}(b_3 - a_{31}x_1 - a_{32}x_2). \tag{4.28}$$

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 (4.26), 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 (4.27) 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 с помощью выражения (4.28) первое

приближение для x_3:

$$x_3^{(1)} = \frac{1}{a_{33}}\left(b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}\right).$$

На этом заканчивается первая итерация решения системы (4.26)—(4.28). 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 представить в виде

$$x_1^{(k)} = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)}\right),$$

$$x_2^{(k)} = \frac{1}{a_{22}}\left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k-1)}\right),$$

$$x_3^{(k)} = \frac{1}{a_{33}}\left(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}\right).$$

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: $x_1 = 1$, $x_2 = 1$, $x_3 = 1$.

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

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(b_i - a_{i1}x_1^{(k)} - \ldots - a_{i,i-1}x_{i-1}^{(k)} - \right.$$
$$\left. - a_{i,i+1}x_{i+1}^{(k-1)} - \ldots - a_{in}x_n^{(k-1)}\right), \quad i = 1, 2, \ldots, n. \qquad (4.29)$$

Итерационный процесс продолжается до тех пор, пока все значения $x_i^{(k)}$ не станут близкими к $x_i^{(k-1)}$. Близость этих значений можно характеризовать максимальной абсолютной величиной их разности δ. Тогда при заданной допустимой погрешности $\varepsilon > 0$ критерий окончания итерационного процесса можно записать в виде

$$\delta = \max_{1\leqslant i\leqslant n} \left| x_i^{(k)} - x_i^{(k-1)} \right| < \varepsilon. \qquad (4.30)$$

Это критерий по абсолютным отклонениям. Можно заменить его критерием по относительным разностям, т. е. условие окончания итерационного процесса записать в виде (при $|x_i| \gg 1$)

$$\max_{1\leqslant i\leqslant n} \left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^k} \right| < \varepsilon. \qquad (4.31)$$

При выполнении условия (4.30) или (4.31) итерационный процесс Гаусса — Зейделя называется *сходящимся*. В этом случае максимальные разности δ между значениями переменных в двух последовательных итерациях

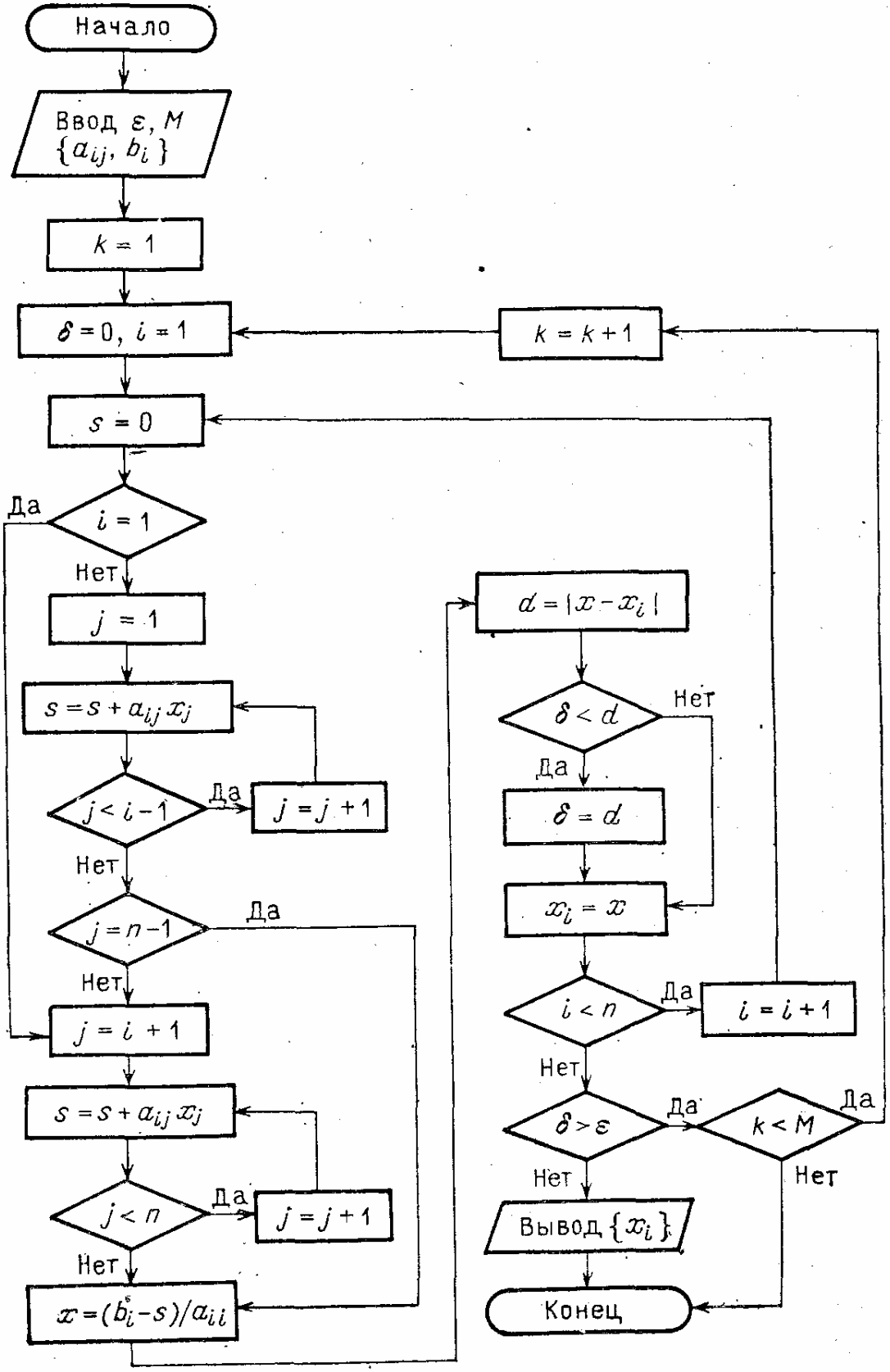


Рис. 20. Блок-схема метода Гаусса — Зейделя

убывают, а сами эти значения стремятся к решению системы уравнений.

Для сходимости итерационного процесса достаточно, чтобы модули диагональных коэффициентов для каждого уравнения системы были не меньше сумм модулей всех остальных коэффициентов:

$$|a_{ii}| \geqslant \sum_{i \neq j} |a_{ij}|, \quad i = 1, 2, \ldots, n. \tag{4.32}$$

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).

Блок-схема алгоритма решения системы n линейных уравнений методом Гаусса — Зейделя представлена на рис. 20. В качестве исходных данных вводятся коэффициенты и правые части уравнений системы, погрешность ε, допустимое число итераций M, а также начальные приближения переменных x_i $(i = 1, 2, \ldots, n)$. Отметим, что начальные приближения можно не вводить в ЭВМ, а полагать их равными некоторым значениям (например, нулю).

Для удобства чтения блок-схемы объясним некоторые обозначения: k — порядковый номер итерации; i — номер уравнения, а также переменного, которое вычисляется в данном цикле; j — номер члена вида $a_{ij}x_j^{(k)}$ в правой части соотношения (4.29). Итерации прекращаются либо после выполнения условия (4.30), либо при $k = M$. В последнем случае итерации не сходятся, и после M итераций счет прекращается без выдачи результатов. Можно предусмотреть в этом случае также и вывод на печать некоторой поясняющей информации.

§ 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

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}. \qquad (4.33)$$

Характеристической матрицей C данной матрицы A называется матрица вида

$$C = A - \lambda E = \begin{bmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{bmatrix}. \qquad (4.34)$$

Здесь λ — *собственное значение*, E — единичная матрица.

Определитель матрицы C является многочленом n-й степени относительно λ:

$$\det C = c_0\lambda^n + c_1\lambda^{n-1} + \dots + c_{n-1}\lambda + c_n, \qquad (4.35)$$

называемым *характеристическим многочленом.* Корни этого многочлена являются собственными значениями матрицы A.

Вектор $X = \{x_1, x_2, \dots, x_n\}$, соответствующий некоторому собственному значению λ и удовлетворяющий системе уравнений

$$AX = \lambda X, \qquad (4.36)$$

называется *собственным вектором* матрицы A.

Поскольку при умножении собственного вектора на скаляр он остается собственным вектором той же матрицы, то 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.

Если перейти к координатной форме записи вектора X, то с учетом (4.33) систему (4.36) можно записать в виде

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

Иногда эту систему уравнений удобно представить в другом виде:

$$\begin{aligned}
&(a_{11}-\lambda)x_1+a_{12}x_2+\ldots+a_{1n}x_n=0,\\
&a_{21}x_1+(a_{22}-\lambda)x_2+\ldots+a_{2n}x_n=0,\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&a_{n1}x_1+a_{n2}x_2+\ldots+(a_{nn}-\lambda)x_n=0.
\end{aligned}$$

Это уже однородная система n линейных уравнений с n неизвестными. Она имеет ненулевые решения лишь тогда, когда ее определитель равен нулю: $\det(A-\lambda E)=0$, причем решение не единственно, поскольку обычно одно уравнение является следствием остальных.

На практике обычно при нахождении собственных векторов матрицы одну из его компонент полагают равной некоторому числу, например $x_1=1$. Остальные компоненты 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

$$A=\begin{bmatrix}3 & 1\\ 2 & 4\end{bmatrix}.$$

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 X_1, X_2, соответствующих собственным значениям λ_1, λ_2, составим системы уравнений типа (4.36) для каждого из них.

При $\lambda_1=2$ получим

$$\begin{bmatrix}3 & 1\\ 2 & 4\end{bmatrix}\cdot\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2x_1\\ 2x_2\end{bmatrix},$$

или в виде системы уравнений:

$$\begin{aligned}
3x_1+x_2&=2x_1,\\
2x_1+4x_2&=2x_2.
\end{aligned}$$

Записывая полученную систему в виде

$$x_1 + x_2 = 0,$$
$$x_1 + x_2 = 0,$$

замечаем, что уравнения линейно зависимы (даже совпадают). Поэтому оставляем лишь одно из них.

Полагаем $x_1 = 1$. Тогда $x_2 = -x_1 = -1$, и собственный вектор, соответствующий собственному значению $\lambda_1 = 2$, имеет вид $X_1 = \{1, -1\}$ или $X_1 = e_1 - e_2$, где e_1, e_2 — единичные орты выбранной базисной системы.

Аналогично находим второй собственный вектор, соответствующий собственному значению $\lambda_2 = 5$. Опуская комментарии, получаем

$$\begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5x_1 \\ 5x_2 \end{bmatrix};$$

$$3x_1 + x_2 = 5x_1, \quad -2x_1 + x_2 = 0,$$
$$2x_1 + 4x_2 = 5x_2; \quad 2x_1 - x_2 = 0.$$

Отсюда $x_1 = 1$, $x_2 = 2$, $X_2 = e_1 + 2e_2$.

Вектор X_1 нормирован; нормируем также вектор X_2, разделив его компоненты на наибольшую из них. Получим $X_2 = 0.5e_1 + e_2$. Можно также привести векторы к единичной длине, разделив их компоненты на значения модулей векторов. В этом случае

$$X_1 = \frac{1}{\sqrt{2}}(e_1 - e_2), \quad X_2 = \frac{1}{\sqrt{5}}(e_1 + 2e_2).$$

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 (4.35). 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

$$B = P^{-1}AP, \tag{4.37}$$

то их собственные значения совпадают (здесь P — некоторая матрица).

Преобразование подобия (4.37) 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

$$\begin{bmatrix} a'_{11} & a'_{12} & \dots & a'_{1n} \\ & a'_{22} & \dots & a'_{2n} \\ & & \dots & \\ 0 & & & a'_{nn} \end{bmatrix}.$$

Тогда матрица (4.34)также имела бы треугольный вид. Как известно, определитель треугольной матрицы равен произведению ее диагональных элементов, поэтому характеристический многочлен (4.35) в этом случае имеет вид

$$\det C = (a'_{11} - \lambda)(a'_{22} - \lambda) \dots (a'_{nn} - \lambda). \tag{4.38}$$

Собственные значения матрицы, равные корням этого многочлена, можно сразу получить:

$$\lambda_1 = a'_{11}, \lambda_2 = a'_{22}, \dots, \lambda_n = a'_{nn}. \tag{4.39}$$

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 симметричную 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, и поворот производится в плоскости, проходящей через эти оси; остальные оси координат на данном шаге неподвижны. *Матрица вращения* при этом имеет вид

$$P_{ij} = \begin{bmatrix} 1 & & & & & & 0 \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & p_{ii} & \cdots & p_{ij} & \\ & & & \vdots & \ddots & \vdots & \\ & & & p_{ji} & \cdots & p_{jj} & \\ & & & & & & \ddots \\ & & & & & & & 1 \\ 0 & & & & & & & & 1 \end{bmatrix}, \qquad (4.40)$$

$$p_{ii} = p_{jj} = p, \quad p_{ij} = -p_{ji} = q, \quad p = \cos\varphi, \quad q = -\sin\varphi.$$

Здесь мы рассматриваем матрицы с вещественными элементами. В случае комплексных векторов для использования этого метода нужно изменить формулы (4.40).

Для осуществления преобразования подобия (4.37) необходимо найти обратную матрицу P_{ij}^{-1}. Можно показать, что она равна в рассматриваемом случае транспонированной матрице P_{ij}^{T}, т. е. 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

$$A_k = P_{ij}^T A_{k-1} P_{ij}, \quad k = 1, 2, \ldots \tag{4.41}$$

Рассмотрим первый шаг преобразования. Сначала вычисляется произведение матриц $B = A_0 P_{ij}$ (здесь A_0 — исходная матрица A). 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.

$$\begin{gathered} b_{ki} = a_{ki}^{(0)} p + a_{kj}^{(0)} q, \quad b_{kj} = -a_{ki}^{(0)} q + a_{kj}^{(0)} p, \\ b_{kl} = a_{kl}^{(0)}, \quad l \neq i, j, \quad k = 1, 2, \ldots, n. \end{gathered} \tag{4.42}$$

Затем находится преобразованная матрица $A_1 = P_{ij}^T B$. Элементы полученной матрицы отличаются от элементов матрицы B только i-й и j-й строками. Они связаны соотношениями

$$a_{ik}^{(1)} = b_{ik} p + b_{jk} q, \quad a_{jk}^{(1)} = -b_{ik} p + b_{jk} q, \quad k = 1, 2, \ldots, n, \tag{4.43}$$

$$a_{kl}^{(1)} = b_{kl}, \quad k \neq i, j, \quad l = 1, 2, \ldots, n.$$

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, 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).

Рассмотрим сначала первый шаг данного метода, состоящий в обращении в нуль элементов, стоящих на местах элементов a_{13}, a_{31}. Для этого 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 $k = 1$, $i = 2$, $j = 3$, 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:

$$a_{13}p - a_{12}q = 0,$$
$$p^2 + q^2 = 1.$$

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 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

$$p = \frac{a_{i-1,i}^{(k-1)}}{\sqrt{\left(a_{i-1,i}^{(k-1)}\right)^2 + \left(a_{i-1,j}^{(k-1)}\right)^2}}, \quad q = \frac{a_{i-1,j}^{(k-1)}}{\sqrt{\left(a_{i-1,i}^{(k-1)}\right)^2 + \left(a_{i-1,j}^{(k-1)}\right)^2}}. \tag{4.45}$$

Учитывая найденные значения параметров p, q, можно по формулам (4.42), (4.43) найти элементы преобразованной матрицы. Процесс вычислений объясним с ис-

пользованием схематического изображения матрицы (рис. 21). Точками отмечены элементы матрицы. Вертикальными линиями показаны столбцы с номерами i, j, горизонтальными — строки с теми же номерами. Наклонные линии указывают три диагонали матрицы, элементы на которых после окончания расчета отличны от нуля; все остальные — нули. На рассматриваемом шаге матрица преобразуется таким образом, чтобы отмеченные крестиками элементы обратились в нуль.

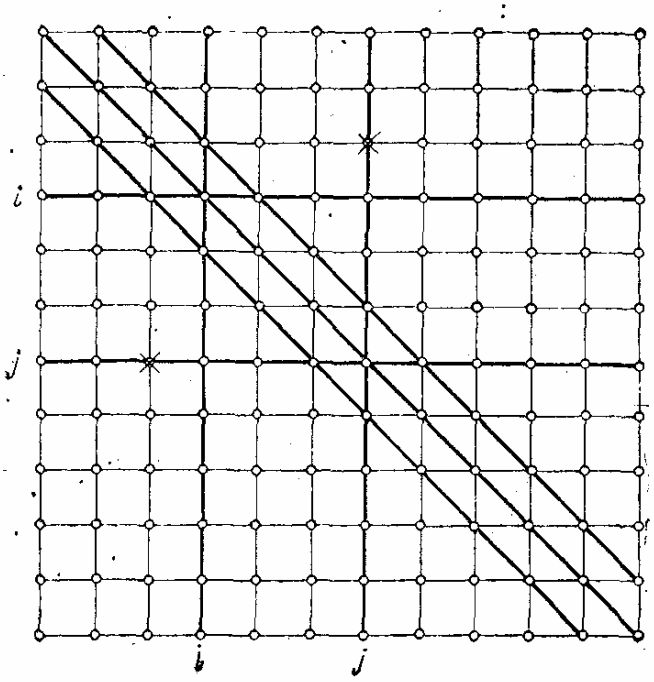


Рис. 21.

Алгоритм решения задачи нужно построить таким образом, чтобы все элементы по одну сторону от трех диагоналей обратились в нуль; тогда симметрично расположенные элементы также станут нулевыми. Преобразование подобия на каждом шаге требует пересчета всех элементов отмеченных столбцов и строк. Учитывая симметрию, можно вычислить лишь все элементы столбцов, а элементы строк получаются из условий симметрии. Исключение составляют лишь элементы, расположенные на пересечениях этих строк и столбцов. Они изменяются на каждом из двух этапов выполняемого шага.

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. Блок-схема приведения симметрической матрицы к трехдиагональному виду с помощью прямого метода вращений представлена на рис. 22.

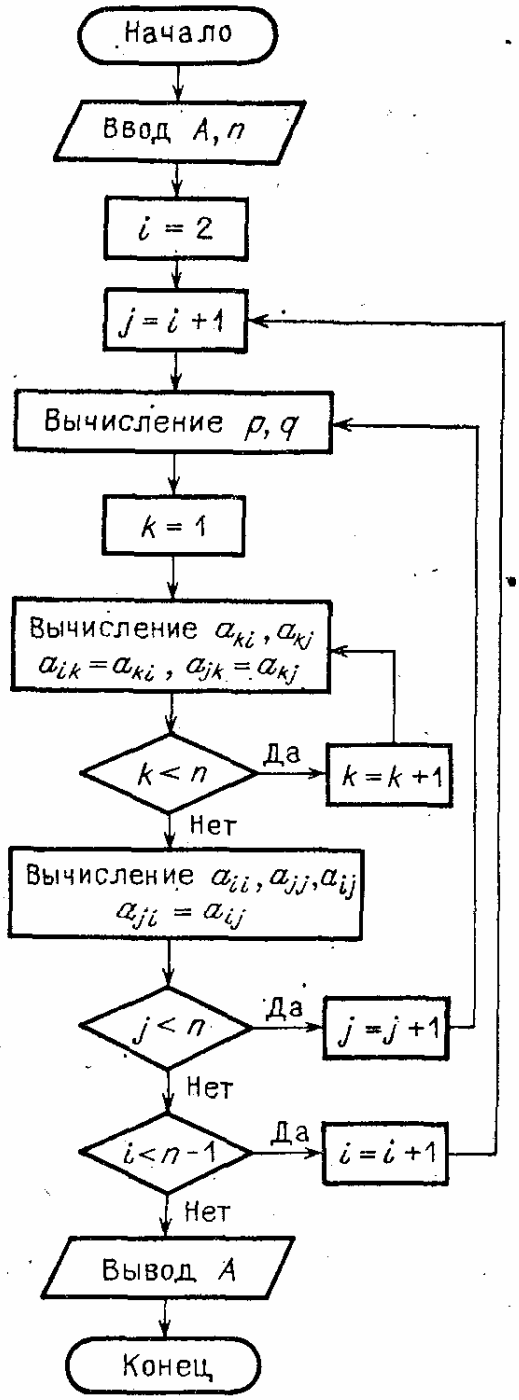


Рис. 22. Блок-схема метода вращений

Собственные значения полученной трехдиагональной матрицы будут также собственными значениями исходной матрицы. Собственные векторы X_i исходной матрицы не равны непосредственно собственным векторам Y_i трехдиагональной матрицы, а вычисляются с помощью соотношений

$$X_i = P_{23}P_{24}\ldots P_{n-1,\,n}Y_i. \tag{4.46}$$

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

$$A = \begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & 0 \\ \cdot & \cdot & \cdot & \cdot & & \\ & & a_{n-1} & b_{n-1} & c_{n-1} & \\ 0 & & & a_n & b_n & \end{bmatrix}. \tag{4.47}$$

Здесь элементы $b_1, b_2, \ldots, b_n$ расположены вдоль главной диагонали, $c_1, c_2, \ldots, c_{n-1}$ — над ней; $a_2, a_3, \ldots, a_n$ — под ней.

Для нахождения собственных значений нужно приравнять нулю определитель $D_n(\lambda) = \det(A - \lambda E)$, или

$$D_n(\lambda) = \begin{vmatrix} b_1 - \lambda & c_1 & & & 0 \\ a_2 & b_2 - \lambda & c_2 & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & a_{n-1} & b_{n-1} - \lambda & c_{n-1} \\ 0 & & & a_n & b_n - \lambda \end{vmatrix} = 0. \tag{4.48}$$

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 (4.48) по элементам последней строки, в которой всего два ненулевых элемента. Получим

$$D_n(\lambda) = (b_n - \lambda) D_{n-1}(\lambda) - a_n M_{n-1}(\lambda), \tag{4.49}$$

$$M_{n-1}(\lambda) = \begin{vmatrix} b_1 - \lambda & c_1 & & 0 \\ a_2 & b_2 - \lambda & c_2 & \\ \cdots & \cdots & \cdots & \cdots \\ 0 & & a_{n-1} & c_{n-1} \end{vmatrix}.$$

Поскольку минор $M_{n-1}(\lambda)$ содержит в последнем столбце лишь один 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 (4.49), получаем рекуррентные соотношения, выражающие минор высшего порядка через миноры двух низших порядков:

$$D_n(\lambda) = (b_n - \lambda) D_{n-1}(\lambda) - a_n c_{n-1} D_{n-2}(\lambda). \tag{4.50}$$

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 (4.50) при $n = 2$:

$$D_2(\lambda) = (b_2 - \lambda) D_1(\lambda) - a_2 c_1 D_0(\lambda) = (b_2 - \lambda)(b_1 - \lambda) - a_2 c_1. \tag{4.51}$$

Вычисляя минор второго порядка определителя (4.48), убеждаемся в справедливости выражения (4.51). Таким

образом, используя рекуррентные соотношения (4.50), можно найти выражение для характеристического многочлена $D_n(\lambda)$. Вычисляя корни этого многочлена, получаем собственные значения $\lambda_1, \lambda_2, \ldots, \lambda_n$ трехдиагональной матрицы (4.47).

Будем считать, что собственные значения $\lambda_1, \lambda_2, \ldots, \lambda_n$ матрицы (4.47) вычислены. Найдем соответствующие им собственные векторы. В соответствии с определением (4.36) собственный вектор для любого собственного значения λ находится из системы уравнений $AX = \lambda X$, или

$$(A - \lambda E)X = 0. \tag{4.52}$$

Перейдем от матричной формы записи этой системы к развернутой (A — матрица вида (4.47), $X = \{x_1, x_2, \ldots, x_n\}$):

$$\begin{aligned}
&(b_1 - \lambda)x_1 + c_1x_2 = 0,\\
&a_2x_1 + (b_2 - \lambda)x_2 + c_2x_3 = 0,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{n-1}x_{n-2} + (b_{n-1} - \lambda)x_{n-1} + c_{n-1}x_n = 0,\\
&a_nx_{n-1} + (b_n - \lambda)x_n = 0.
\end{aligned} \tag{4.53}$$

Матрица системы (4.53) вырожденная, поскольку ее определитель (4.48) равен нулю. Можно показать, что последнее уравнение системы (4.53) является следствием остальных уравнений, если $c_i \neq 0$ $(i = 1, 2, \ldots, n-1)$. Действительно, если отбросить первый столбец и последнюю строку в матрице A, то вместо (4.48) получится определитель вида

$$\begin{vmatrix} c_1 & & & 0 \\ b_2 - \lambda & c_2 & & \\ \cdot\cdot\cdot & \cdot\cdot\cdot & \cdot\cdot\cdot & \cdot\cdot\cdot \\ 0 & a_{n-1} & b_{n-1} - \lambda & c_{n-1} \end{vmatrix} = c_1c_2\ldots c_{n-1} \neq 0. \tag{4.54}$$

Следовательно, все строки с первой по $n-1$-ю линейно независимы. Отбрасывая последнее уравнение системы (4.53), записываем ее в виде

$$\begin{aligned}
&c_1x_2 = -(b_1 - \lambda)x_1,\\
&(b_2 - \lambda)x_2 + c_2x_3 = -a_2x_1,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{n-1}x_{n-2} + (b_{n-1} - \lambda)x_{n-1} + c_{n-1}x_n = 0.
\end{aligned} \tag{4.55}$$

Полагая компоненту x_1 равной любому ненулевому значению, можно из системы (4.55) 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 (4.54) этой системы отличен от нуля, то она имеет единственное решение. Описанным способом могут быть найдены собственные векторы, соответствующие всем собственным значениям $\lambda_1, \lambda_2, \ldots, \lambda_n$ трехдиагональной матрицы (4.47).

Если трехдиагональная 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 (4.46). При этом вычислять произведения матриц $P_{23}, P_{24}, \ldots, P_{n-1, n}$ на собственные векторы трехдиагональной матрицы нецелесообразно, поскольку при умножении матрицы P_{ij} на вектор X изменяются только две его компоненты: x_i, 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 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.

Итерационный процесс строится на основании применения методов итераций к решению системы уравнений

$$\lambda X = AX. \tag{4.56}$$

Используем метод простой итерации. Пусть $X^{(0)}$ — начальное приближение собственного вектора X, причем собственные векторы на каждой итерации нормированы,

Итерационный процесс запишется в виде

$$\lambda^{(k+1)}X^{(k+1)} = AX^{(k)}, \quad k = 0, 1, \ldots \qquad (4.57)$$

Подставляя в правую часть этой системы вектор $X^{(0)}$, получаем после его умножения слева на матрицу A некоторый вектор $Y^{(1)}$. После нормировки этого вектора он представится в виде $Y^{(1)} = \lambda^{(1)}X^{(1)}$, где $\lambda^{(1)}$ — постоянная, $X^{(1)}$ — нормированный вектор. Теперь нужно $X^{(1)}$ снова подставить в правую часть (4.57) и найти новые приближения $\lambda^{(2)}$ и $X^{(2)}$. Итерационный процесс продолжается до установления постоянных значений λ и X. При этом найденное число λ — наибольшее по модулю собственное значение данной матрицы A, а X — соответствующий ему собственный вектор.

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 (4.56) можно использовать и другие итерационные методы. В частности, метод Ньютона дает лучшую сходимость, если удачно выбрано начальное приближение $X^{(0)}$. В этом случае бывает достаточно нескольких итераций.

В некоторых задачах нужно искать не наибольшее, а наименьшее собственное значение матрицы A. В этом случае можно умножить систему (4.56) на обратную матрицу A^{-1}:

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

Отсюда, деля обе части этого равенства на λ и учитывая, что $A^{-1}A = E$, получаем

$$\frac{1}{\lambda}X = A^{-1}X. \qquad (4.58)$$

Эта задача отличается от ранее рассматриваемой тем, что здесь будет вычисляться наибольшее собственное значение $1/\lambda$, что будет достигнуто при наименьшем λ. Следовательно, рассмотренный выше итерационный процесс может быть использован также для нахождения наименьшего собственного значения обратной матрицы (собственные значения матриц A и A^{-1} обратны друг другу).

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. Составить блок-схему вычисления определителя данной матрицы.

3. Используя метод Гаусса, решить следующие системы уравнений с погрешностью 10^{-4}:

а) $1.17x_1 + 0.53x_2 - 0.84x_3 = 1.15,$
$0.64x_1 - 0.72x_2 - 0.43x_3 = 0.15,$
$0.32x_1 + 0.43x_2 - 0.93x_3 = -0.48;$

б) $1.20x_1 - 0.20x_2 + 0.30x_3 = -0.60,$
$-0.20x_1 + 1.60x_2 - 0.10x_3 = 0.30,$
$-0.30x_1 + 0.10x_2 - 1.50x_3 = 0.40.$

4. Составить блок-схему вычисления обратной матрицы по методу Гаусса. Блок прямого хода можно считать заданным (см. рис. 16).

5. С помощью метода прогонки решить систему уравнений

$$\begin{aligned} 2x_1 + 2x_2 &= 1,\\ -x_1 + x_2 - 0.5x_3 &= 0,\\ x_2 - 3x_3 - x_4 &= 2,\\ x_3 + 2x_4 &= 2. \end{aligned}$$

6. Решить методом Гаусса — Зейделя с погрешностью 10^{-3} системы уравнений:

а) $5.6x_1 + 2.7x_2 - 1.7x_3 = 1.9,$
$3.4x_1 - 3.6x_2 - 6.7x_3 = -2.4,$
$0.8x_1 + 1.3x_2 + 3.7x_3 = 1.2;$

б) $7.1x_1 + 6.8x_2 + 6.1x_3 = 7.0,$
$5.0x_1 + 4.8x_2 + 5.3x_3 = 6.1,$
$8.2x_1 + 7.8x_2 + 7.1x_3 = 5.8.$

7. Найти собственные значения и собственные векторы матриц

$$A = \begin{bmatrix} 3 & -1 \\ 2 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 2 \\ 0 & 2 & 3 \end{bmatrix}.$$

8. Составить алгоритм приведения матрицы четвертого порядка к трехдиагональному виду и решения полной проблемы собственных значений.

9. Составить блок-схему 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. Алгоритм нахождения корня уравнения с помощью итерационного метода состоит из двух этапов: а) отыскания приближенного значения корня или содержащего его отрезка; б) уточнения приближенного значения до некоторой заданной степени точности.

Приближенное значение корня (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. $x_0=(a+b)/2$.

Итерационный процесс состоит в последовательном уточнении начального приближения x_0. Каждый такой шаг называется *итерацией*. В результате итераций находится последовательность приближенных значений корня $x_1, x_2, \ldots, x_n$. Если эти значения с ростом n приближаются к истинному значению корня, то говорят, что итерационный процесс *сходится*.

Ниже рассматриваются некоторые итерационные методы решения трансцендентных уравнений. Они могут использоваться также и для нахождения корней алгебраических уравнений, некоторые 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 $[a, b]$, в котором расположено искомое значение корня $x=c$, т. е. $a<c<b$. В качестве начального приближения корня c_0 принимаем середину этого отрезка, т. е. $c_0=(a+b)/2$. 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 них, на концах которого $F(x)$ принимает значения разных знаков, содержит искомый корень; поэтому его принимаем в качестве нового отрезка. Вторую половину отрезка $[a, b]$, на которой знак $F(x)$ не меняется, отбрасываем. В качестве первой итерации корня принимаем середину нового отрезка и т. д. Таким образом, после каждой итерации отрезок, на котором расположен корень, уменьшается вдвое, т. е. после n итераций он сокращается в 2^n раз.

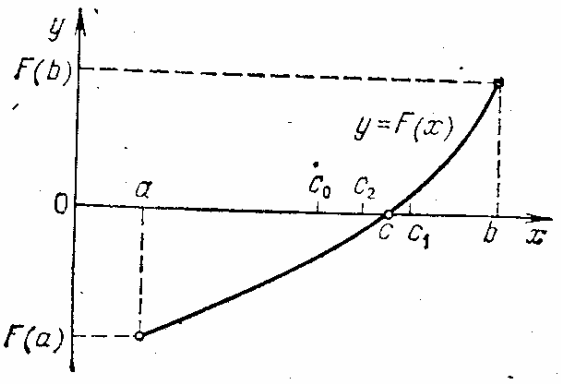


Рис. 23. Метод деления отрезка пополам

Пусть для определенности $F(a)<0, F(b)>0$ (рис. 23). В качестве начального приближения корня примем $c_0=$

$=(a+b)/2$. Поскольку в рассматриваемом случае $F(c_0)<0$, то $c_0<c<b$, и рассматриваем только отрезок $[c_0, b]$. Следующее приближение: $c_1=(c_0+b)/2$. При этом отрезок $[c_1, b]$ отбрасываем, поскольку $F(c_1)>0$ и $F(b)>0$, т. е. $c_0<c<c_1$. Аналогично находим другие приближения: $c_2=(c_0+c_1)/2$ и т. д.

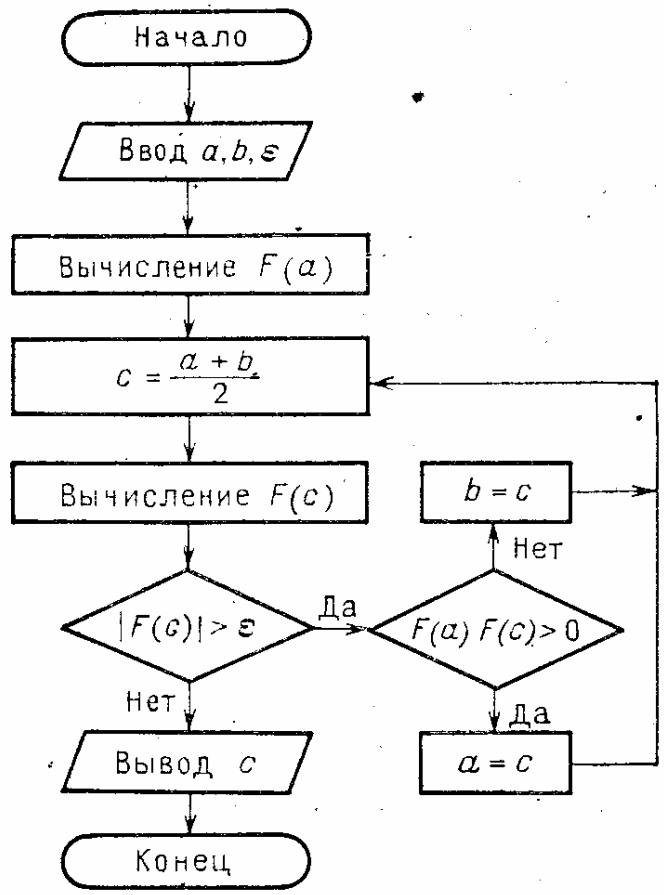


Рис. 24. Блок-схема метода деления отрезка пополам

Итерационный процесс продолжаем до тех пор, пока значение функции $F(x)$ после n-й итерации не станет меньшим по модулю некоторого заданного малого числа ε, т. е. $|F(c_n)|<\varepsilon$. Можно также оценивать длину полученного отрезка: если она становится меньше допустимой погрешности, то счет прекращается.

На рис. 24 представлена блок-схема 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.

Метод деления отрезка пополам довольно медленный, однако он всегда сходится, т. е. при его использовании решение получается всегда, причем с заданной точностью (разумеется, в рамках разрядности ЭВМ). Требуемое обычно большее число итераций по сравнению с некоторыми другими методами не является препятствием к применению этого метода, если каждое вычисление значения функции $F(x)$ несложно.

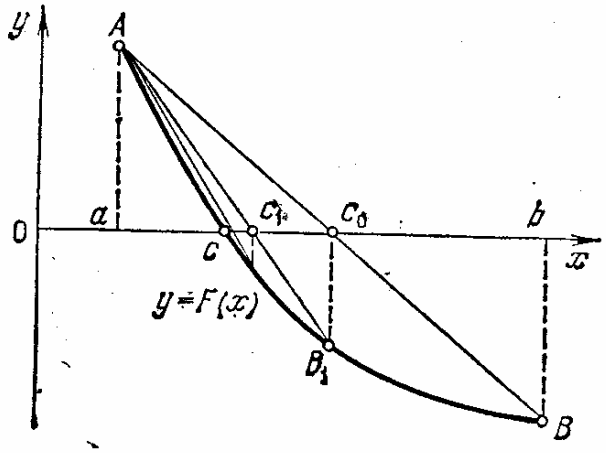


Рис. 25. Метод хорд

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. 25). В данном методе процесс итераций состоит в том, что в качестве приближений к 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

$$c_0 = a - \frac{b - a}{F(b) - F(a)} F(a). \qquad (5.2)$$

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. Итерационный процесс продолжается до тех пор, пока значение $F(c_n)$ не станет по модулю меньше заданного числа ε.

Блок-схема метода хорд аналогична приведенной на рис. 24 для метода деления отрезка пополам с той лишь разницей, что вместо вычисления приближения корня по формуле $c = (a + b)/2$ нужно использовать формулу (5.2).

Кроме того, в блок-схему необходимо ввести операторы вычисления значений $F(x)$ на границах новых отрезков. Советуем читателю самостоятельно построить блок-схему метода хорд.

Как видим, алгоритмы метода деления отрезка пополам и метода хорд похожи, однако второй из них в ряде случаев дает более быструю сходимость итерационного процесса. При этом успех его применения, как и в методе деления отрезка пополам, гарантирован.

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 $x = c_k$ 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. 26).

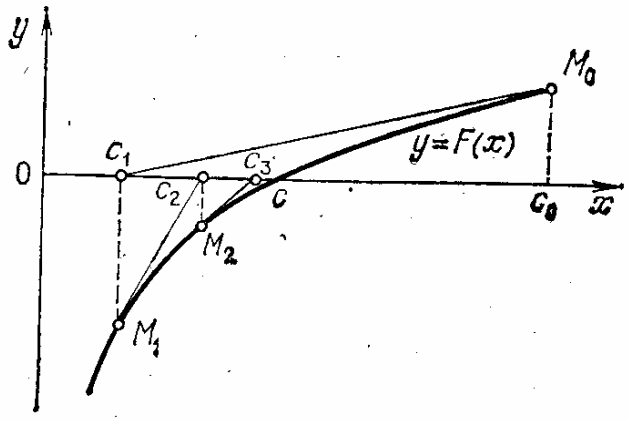


Рис. 26. Метод касательных

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 $n+1$-го приближения имеет вид

$$c_{n+1} = c_n - F(c_n)/F'(c_n). \qquad (5.3)$$

При этом необходимо, чтобы $F'(c_n)$ не равнялась нулю. Для окончания итерационного процесса может быть использовано или условие $|F(c_n)| < \varepsilon$, или условие близости двух последовательных приближений: $|c_{n+1} - c_n| < \varepsilon$.

Из (5.3) 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. *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* $\{c_n\}$ *сходится к* c *при* $n \to \infty$. *При этом для погрешности корня* $\varepsilon_n = c_n - c$ *имеет место соотношение*

$$\lim_{n\to\infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^2} = \frac{F''(c)}{2F'(c)}.$$

Фактически это означает, что на каждой итерации погрешность возводится в квадрат, т. е. число верных знаков корня удваивается. Если

$$\frac{F''(c)}{2F'(c)} \sim 1,$$

то легко показать, что при $|\varepsilon_n| < 0.5$ после пяти-шести итераций погрешность станет величиной порядка 2^{-64}. Это наименьшее возможное значение погрешности при вычислениях на современных ЭВМ даже с удвоенной точностью. Заметим, что для получения столь малой погрешности в методе деления отрезка пополам потребовалось бы более 50 итераций.

Трудность в применении метода Ньютона состоит в выборе начального приближения, которое должно находиться в окрестности D. Поэтому 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

$$x = f(x). \tag{5.4}$$

Пусть известно начальное приближение корня $x = c_0$. Подставляя это значение в правую часть уравнения (5.4), получаем новое приближение:

$$c_1 = f(c_0).$$

Далее, подставляя каждый раз новое значение корня в (5.4), получаем последовательность значений

$$c_{n+1} = f(c_n), \quad n = 1, 2, \ldots$$

Итерационный процесс прекращается, если результаты двух последовательных итераций близки: $|c_{n+1} - c_n| < \varepsilon$. Достаточным условием сходимости метода простой итерации является условие $|f'(c_n)| < 1$.

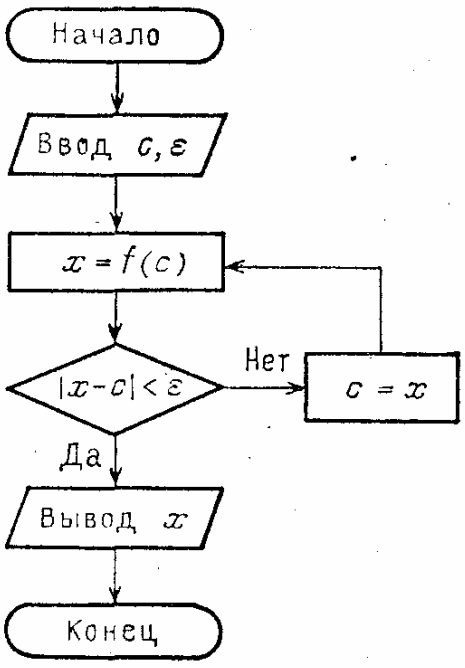


Рис. 27. Блок-схема метода простой итерации

На рис. 27 представлена блок-схема процесса решения нелинейного уравнения (5.4) методом простой итерации. Здесь c — начальное приближение корня, а в дальнейшем — результат предыдущей итерации, x — значение корня после каждой 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. 20).

§ 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 (см. 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 известные из курса алгебры некоторые свойства алгебраических уравнений с действительными коэффициентами вида

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0. \qquad (5.5)$$

1. Уравнение степени n имеет всего n корней, 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 $x = c - id$.

3. Число положительных действительных корней меньше или равно числу перемен знаков в последовательности коэффициентов $a_0, a_1, \ldots, a_n$. Заменяя x на $-x$ в уравнении (5.5), таким же способом можно оценить число отрицательных корней.

Одним из способов решения уравнения (5.5) 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 (5.5). В соответствии с формулой (5.3) итерационный процесс для нахождения корня нелинейного уравнения (5.5) имеет вид

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

$$F(x) = a_0 + a_1 x + \ldots + a_n x^n, \quad F'(x) = a_1 + 2a_2 x + \ldots + n a_n x^{n-1}.$$

Для вычисления значений многочленов $F(x)$ и $F'(x)$ в точке $x = x_n$ может быть использована схема Горнера (см. гл. 2, § 2, п. 3).

Естественно, при использовании метода Ньютона должны выполняться условия сходимости (см. § 1, п. 4). При их соблюдении в результате 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 ЭВМ, как правило, имеется возможность работать с комплексными числами; поэтому изложенные методы могут быть использованы и для нахождения комплексных корней многочленов. Если в качестве начального приближения x_0 взять комплексное число, то последующие приближения и окончательное значение корня могут оказаться комплексными.

Комплексные корни попарно сопряженные, и при их исключении порядок уравнения уменьшается на два, поскольку оно делится сразу на квадратный трехчлен, т. е.

$$F(x) = (x^2 + px + q)(b_n x^{n-2} + \ldots + b_2) + b_1 x + b_0. \quad (5.6)$$

Линейный остаток $b_1 x + b_0$ равен нулю, если p, q выразить с помощью найденных корней, т. е.

$$p = -2c, \quad q = c^2 + d^2, \quad x = c \pm id.$$

Представление (5.6) 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_1, b_0 равны нулю. Тогда, сравнивая коэффициенты при одинаковых степенях x многочлена $F(x)$ в выражениях (5.5) и (5.6), можно получить (для упрощения выкладок $b_n = a_n = 1$)

$$\begin{aligned} b_{n-1} &= a_{n-1} - p, \\ b_{n-2} &= a_{n-2} - pb_{n-1} - q, \\ &\cdots\cdots\cdots \\ b_2 &= a_2 - pb_3 - qb_4; \\ p &= (a_1 - qb_3)/b_2, \quad q = a_0/b_2. \end{aligned} \quad (5.7)$$

Задаем начальные приближения для p, q, которые используются для вычисления коэффициентов b_{n-1}, b_{n-2}, ..., b_2. Затем из двух последних уравнений системы (5.7) уточняем значения p, q. Итерационный процесс вычисления этих величин продолжается до тех пор, пока их из-

менения в двух последующих итерациях не станут малыми. Таким образом, в методе Лина проводится решение двух линейных уравнений относительно p и q по методу итераций Гаусса — Зейделя.

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, который будет рассмотрен ниже.

§ 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 $x_1, x_2, \ldots, x_n$ требуется решить систему n нелинейных уравнений

$$\begin{aligned} F_1(x_1, x_2, \ldots, x_n)&=0,\\ F_2(x_1, x_2, \ldots, x_n)&=0,\\ \cdots\cdots\cdots&\\ F_n(x_1, x_2, \ldots, x_n)&=0. \end{aligned} \tag{5.8}$$

В отличие от систем линейных уравнений не существует прямых методов решения нелинейных систем общего вида. Лишь в отдельных случаях систему (5.8) 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 два из них — метод простой итерации и метод Ньютона.

2. Метод простой итерации. Систему уравнений (5.8) представим в виде

$$\begin{aligned} x_1&=f_1(x_1, x_2, \ldots, x_n),\\ x_2&=f_2(x_1, x_2, \ldots, x_n),\\ \cdots\cdots\cdots&\\ x_n&=f_n(x_1, x_2, \ldots, x_n). \end{aligned} \tag{5.9}$$

Алгоритм решения этой системы методом простой итерации напоминает метод Гаусса — Зейделя, используемый для решения систем линейных уравнений (см. гл. 4, § 3).

Пусть в результате предыдущей итерации получены значения неизвестных $x_1 = a_1$, $x_2 = a_2$, ..., $x_n = a_n$. Тогда выражения для неизвестных на следующей итерации имеют вид

$$\begin{aligned}
x_1 &= f_1(a_1, a_2, \ldots, a_n),\\
x_2 &= f_2(x_1, a_2, \ldots, a_n),\\
&\cdots\cdots\cdots\\
x_i &= f_i(x_1, \ldots, x_{i-1}, a_i, \ldots, a_n),\\
&\cdots\cdots\cdots\\
x_n &= f_n(x_1, \ldots, x_{n-1}, a_n).
\end{aligned}$$

Итерационный процесс продолжается до тех пор, пока изменения всех неизвестных в двух последовательных итерациях не станут малыми, т. е. абсолютные величины их разностей не станут меньшими заданного малого числа.

При использовании метода простой итерации 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. В случае одного уравнения $F(x) = 0$ алгоритм метода Ньютона был легко получен путем записи уравнения касательной к кривой $y = F(x)$. В основе метода Ньютона для системы уравнений лежит использование разложения функций $F_i(x_1, x_2, \ldots, x_n)$ в ряд Тейлора, причем члены, содержащие вторые (и более высоких порядков) производные, отбрасываются.

Пусть приближенные значения неизвестных системы (5.8) (например, полученные на предыдущей итерации) равны соответственно $a_1, a_2, \ldots, a_n$. Задача состоит в нахождении приращений (поправок) к этим значениям $\Delta x_1, \Delta x_2, \ldots, \Delta x_n$, благодаря которым решение системы (5.8) запишется в виде

$$x_1 = a_1 + \Delta x_1, \quad x_2 = a_2 + \Delta x_2, \ldots, x_n = a_n + \Delta x_n. \quad (5.10)$$

Проведем разложение левых частей уравнений (5.8) с учетом (5.10) в ряд Тейлора, ограничиваясь лишь

линейными членами относительно приращений:

$$F_1(x_1, \ldots, x_n) \approx F_1(a_1, \ldots, a_n) + \frac{\partial F_1}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial F_1}{\partial x_n}\Delta x_n,$$

$$F_2(x_1, \ldots, x_n) \approx F_2(a_1, \ldots, a_n) + \frac{\partial F_2}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial F_2}{\partial x_n}\Delta x_n,$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$F_n(x_1, \ldots, x_n) \approx F_n(a_1, \ldots, a_n) + \frac{\partial F_n}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial F_n}{\partial x_n}\Delta x_n.$$

Поскольку в соответствии с (5.8) левые части этих выражений должны обращаться в нуль, то приравняем нулю и правые части. Получим следующую систему линейных алгебраических уравнений относительно приращений:

$$\begin{aligned}
&\frac{\partial F_1}{\partial x_1}\Delta x_1 + \frac{\partial F_1}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial F_1}{\partial x_n}\Delta x_n = -F_1,\\
&\frac{\partial F_2}{\partial x_1}\Delta x_1 + \frac{\partial F_2}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial F_2}{\partial x_n}\Delta x_n = -F_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\frac{\partial F_n}{\partial x_1}\Delta x_1 + \frac{\partial F_n}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial F_n}{\partial x_n}\Delta x_n = -F_n.
\end{aligned} \tag{5.11}$$

Значения $F_1, F_2, \ldots, F_n$ и их производные вычисляются при $x_1 = a_1, x_2 = a_2, \ldots, x_n = a_n$.

Определителем системы (5.11) является *якобиан*

$$J = \begin{vmatrix} \frac{\partial F_1}{\partial x_1} & \ldots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \ldots & \frac{\partial F_2}{\partial x_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial F_n}{\partial x_1} & \ldots & \frac{\partial F_n}{\partial x_n} \end{vmatrix}.$$

Для существования единственного решения системы (5.11) он должен быть отличным от нуля на каждой итерации.

Таким образом, итерационный процесс решения системы уравнений (5.8) методом Ньютона состоит в определении приращений $\Delta x_1, \Delta x_2, \ldots, \Delta x_n$ к значениям неизвестных на каждой итерации. Счет прекращается, если все приращения становятся малыми по абсолютной вели-

чине: $\max_i |\Delta x_i| < \varepsilon$. В методе Ньютона также важен удачный выбор начального приближения для обеспечения хорошей сходимости. Сходимость ухудшается с увеличением числа уравнений системы.

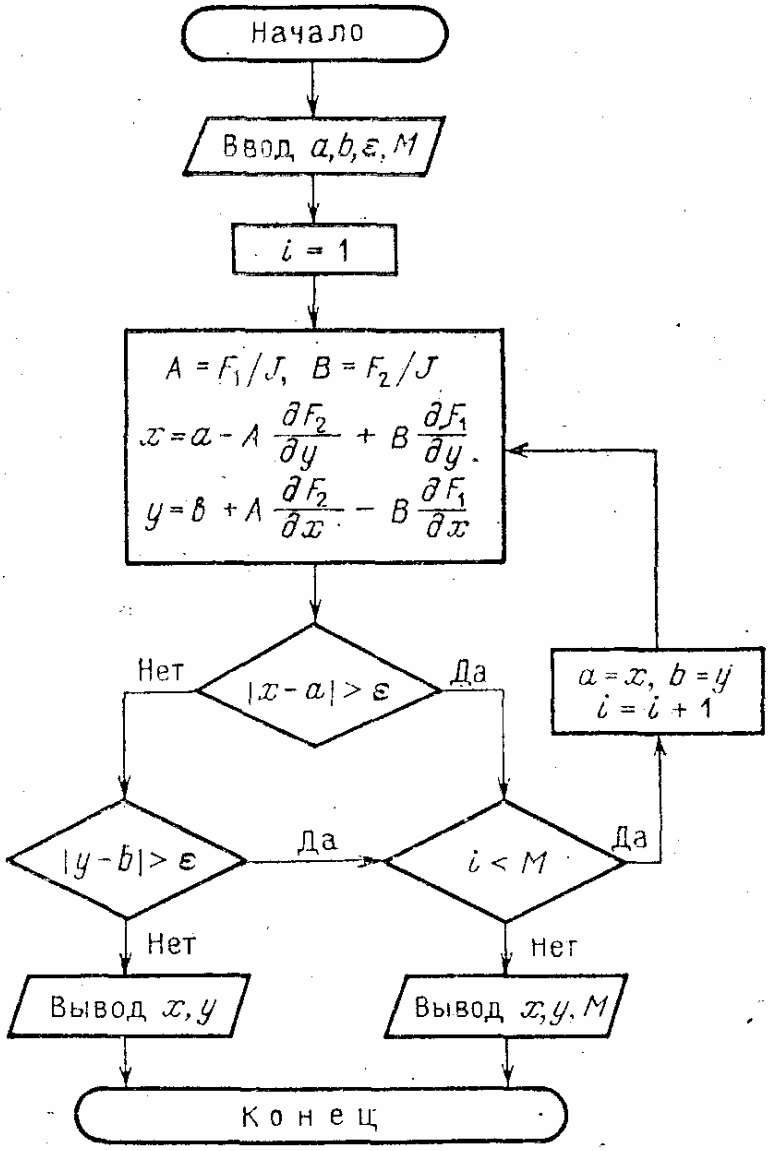


Рис. 28. Блок-схема метода Ньютона для двух систем уравнений

В качестве примера рассмотрим использование метода Ньютона для решения системы двух уравнений

$$F_1(x,\ y) = 0,$$
$$F_2(x,\ y) = 0. \qquad (5.12)$$

Пусть приближенные значения неизвестных равны a, b. Предположим, что якобиан системы (5.12) при $x = a$,

$y = b$ отличен от нуля, т. е.

$$J = \begin{vmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} \end{vmatrix} \neq 0.$$

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 правой части, вычисляются при $x = a$, $y = b$.

Блок-схема метода Ньютона для решения системы двух уравнений изображена на рис. 28. В качестве исходных данных задаются начальные приближения неизвестных a, b, погрешность ε и допустимое число итераций M. Если итерации сойдутся, то выводятся значения x, y; в противном случае происходит вывод x, y, M.

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. Составить блок-схему 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. Построить блок-схему 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. С помощью метода простой итерации найти с погрешностью 10^{-3} хотя бы один корень уравнений: а) $5x - 8 \ln x = 8$; б) $x^2 = \sin x$.

7. Определить глубину погружения деревянного шара радиуса 20 см, плавающего в воде. Плотность дерева 0.75 г/см^3.

8. Найти процентное содержание углекислого газа в реакции $2CO + O_2 \rightleftarrows 2CO_2$, которое определяется уравнением $(p/k^2 - 1)x^3 + 3x - 2 = 0$, где p — давление, k — постоянная равновесия. Принять $p = 1$, $k = 1.648$.

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

$$u = f(x_1, x_2, \ldots, x_n). \qquad (6.1)$$

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 графиком является поверхность.

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

$$\begin{aligned} g_1(x_1, x_2, \ldots, x_n) &= 0, \\ g_2(x_1, x_2, \ldots, x_n) &= 0, \\ \cdots\cdots\cdots\cdots\cdots & \\ g_m(x_1, x_2, \ldots, x_n) &= 0. \end{aligned} \tag{6.2}$$

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

$$\begin{aligned} a_1 \leqslant \varphi_1(x_1, x_2, \ldots, x_n) \leqslant b_1, \\ a_2 \leqslant \varphi_2(x_1, x_2, \ldots, x_n) \leqslant b_2, \\ \cdots\cdots\cdots\cdots\cdots \\ a_k \leqslant \varphi_k(x_1, x_2, \ldots, x_n) \leqslant b_k. \end{aligned} \tag{6.3}$$

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 S должна быть минимальной. Если обозначить через x_1, x_2, x_3 длины ребер контейнера, то задача сведется к минимизации функции

$$S = 2(x_1x_2 + x_2x_3 + x_1x_3). \tag{6.4}$$

Эта функция в данном случае является целевой, а условие $V = 1$ — ограничением-равенством, которое позволяет исключить один параметр:

$$V = x_1 x_2 x_3 = 1, \quad x_3 = \frac{1}{x_1 x_2},$$

$$S = 2\left(x_1 x_2 + \frac{1}{x_1} + \frac{1}{x_2}\right). \tag{6.5}$$

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 (6.5), поскольку ограничение-равенство было использовано для исключения параметра x_3.

Вместе с тем можно рассматриваемую задачу усложнить 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

$$x_1 \geqslant 2. \tag{6.6}$$

Таким образом, мы получили следующую условную задачу оптимизации: минимизируя функцию (6.5) и учитывая ограничение-неравенство (6.6), 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, когда равные экстремальные значения достигаются 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.

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.

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 менее заданного допустимого значения ε, т. е. оптимальное значение проектного параметра должно находиться в интервале неопределенности — отрезке $[x_n, x_{n+1}]$, причем $x_{n+1}-x_n<\varepsilon$.

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 $y_k=f_k(x)$ 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. 29), т. е. интервал неопределенности сузился до длины двух шагов. Если размер интервала недостаточен для удовлетворения заданной погрешности, т. е. $x_{i+1} - x_{i-1} > \varepsilon$, 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.

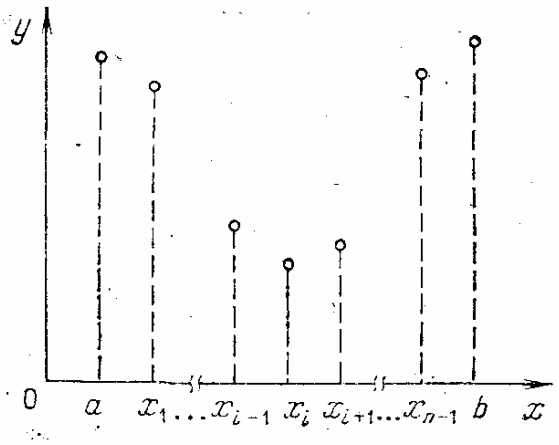


Рис. 29

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 разобъем 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 $[a_0, b_0]$, $[a_1, b_1]$, ..., стягивающихся к точке минимума функции $f(x)$. На каждом шаге, за исключением первого, вычисление значения функции $f(x)$ проводится лишь один раз. Эта точка, называемая *золотым сечением*, выбирается специальным образом.

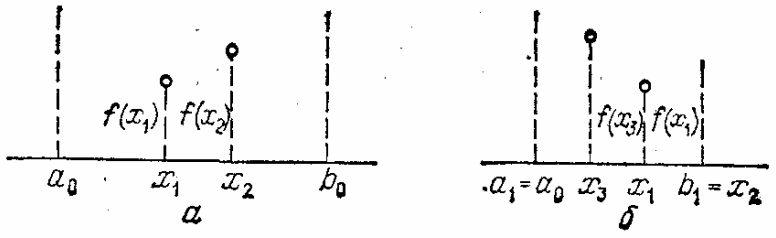


Рис. 30

Поясним сначала идею метода геометрически, а затем выведем необходимые соотношения. На первом шаге процесса оптимизации внутри отрезка $[a_0, b_0]$ (рис. 30, *а*) выбираем две 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. 30, *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 $f(x_3) > f(x_1)$, 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 $[a_n, b_n]$ 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 делит 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, определив отношение l_2/l_1. Преобразуем выражение (6.7) и найдем это значение:

$$l_1^2 = l_2 l, \quad l_1^2 = l_2 (l_1 + l_2),$$

$$l_2^2 + l_1 l_2 - l_1^2 = 0,$$

$$\left(\frac{l_2}{l_1}\right)^2 + \frac{l_2}{l_1} - 1 = 0,$$

$$\frac{l_2}{l_1} = \frac{-1 \pm \sqrt{5}}{2}.$$

Поскольку нас интересует только положительное решение, то

$$\frac{l_2}{l_1} = \frac{l_1}{l} = \frac{-1 + \sqrt{5}}{2} \approx 0.618.$$

Отсюда $l_1 \approx 0.618l$, $l_2 \approx 0.382l$.

Поскольку заранее неизвестно, в какой последовательности (l_1 и l_2 или l_2 и l_1) делить интервал неопределен-

ности, то рассматривают внутренние точки, соответствующие двум этим способам деления. На рис. 30, *а* точки деления x_1 и x_2 выбираются с учетом полученных значений для частей отрезка. В данном случае имеем

$$x_1 - a_0 = b_0 - x_2 = 0.382d_0,$$
$$b_0 - x_1 = x_2 - a_0 = 0.618d_0,$$
$$d_0 = b_0 - a_0.$$

После первого шага оптимизации получается новый интервал неопределенности — отрезок $[a_1, b_1]$ (см. рис. 30, *б*). Можно показать, что точка x_1 делит этот отрезок в требуемом отношении, при этом

$$b_1 - x_1 = 0.382d_1, \quad d_1 = b_1 - a_1.$$

Для этого проведем очевидные преобразования:

$$b_1 - x_1 = x_2 - x_1 = (b_0 - a_0) - (x_1 - a_0) - (b_0 - x_2) =$$
$$= d_0 - 0.382d_0 - 0.382d_0 = 0.236d_0,$$
$$d_1 = x_2 - a_0 = 0.618d_0,$$
$$b_1 - x_1 = 0.236(d_1/0.618) = 0.382d_1.$$

Вторая точка деления x_3 выбирается на таком же расстоянии от левой границы отрезка, т. е. $x_3 - a_1 = 0.382d_1$. И снова интервал неопределенности уменьшается до размера

$$d_2 = b_2 - a_2 = b_1 - x_3 = 0.618d_1 = 0.618^2 d_0.$$

Используя полученные соотношения, можно записать координаты точек деления y и z отрезка $[a_k, b_k]$ на $k+1$-м шаге оптимизации ($y < z$):

$$y = 0.618a_k + 0.382b_k,$$
$$z = 0.382a_k + 0.618b_k. \tag{6.8}$$

При этом длина интервала неопределенности равна

$$d_k = b_k - a_k = 0.618^k d_0. \tag{6.9}$$

Процесс оптимизации заканчивается при выполнении условия $d_k < e$. При этом проектный параметр оптимизации составляет $a_k < x < b_k$. Можно в качестве оптимального значения принять $x = a_k$ (или $x = b_k$, или $x = (a_k + b_k)/2$ и т. п.).

На рис. 31 представлена блок-схема процесса одномерной оптимизации методом золотого сечения. Здесь y, z — точки деления отрезка $[a, b]$, причем $y < z$. В результате выполнения алгоритма выдается оптимальное значение

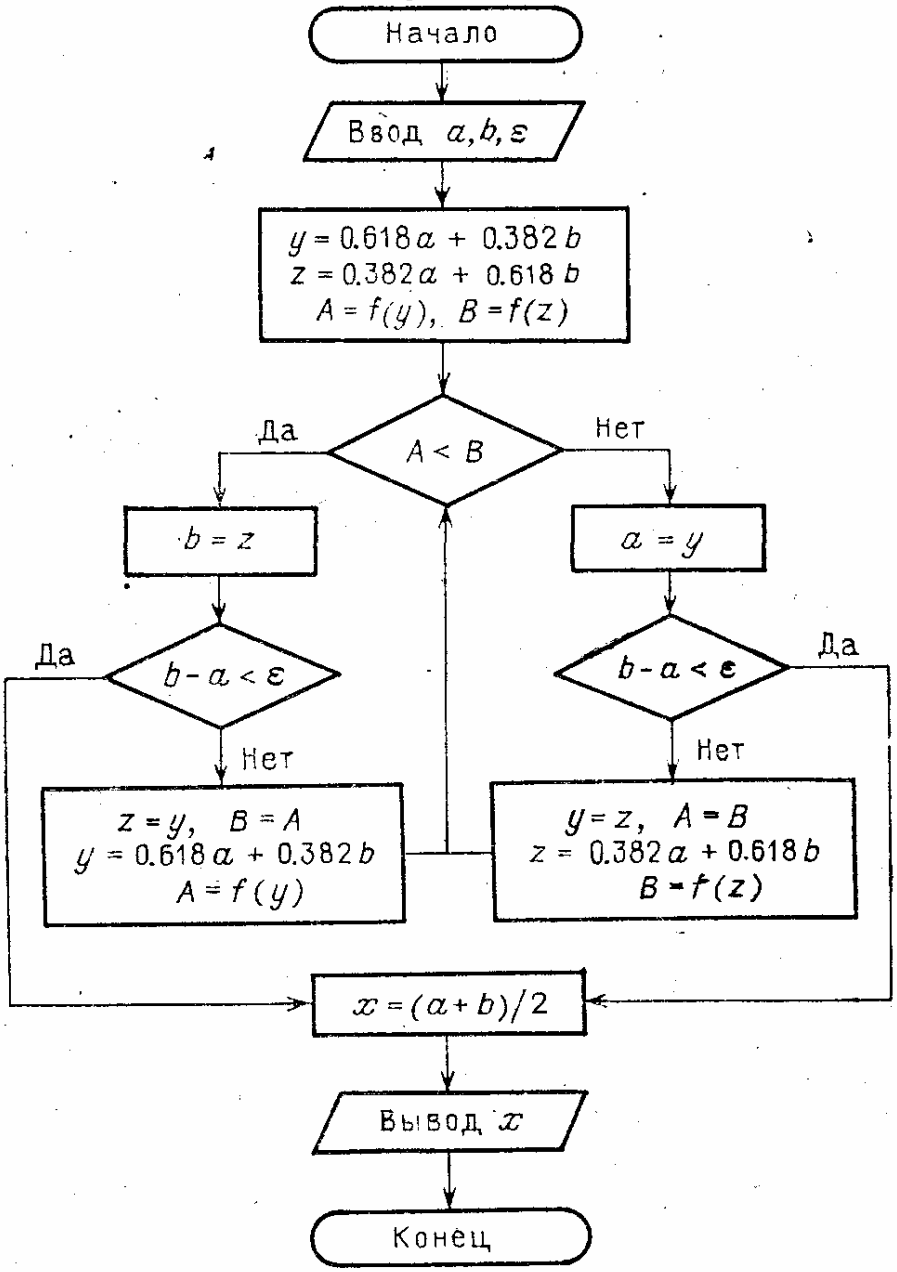


Рис. 31. Блок-схема метода золотого сечения

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 $f(v) = 24 - \frac{2}{3}v + \frac{1}{30}v^2$ (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,

$$f'(v) = -\frac{2}{3} + \frac{2v}{30} = 0, \quad v = 10 \text{ км/ч}.$$

Проиллюстрируем на этой простейшей задаче метод золотого сечения. Первоначально границы интервала неопределенности примем равными $a = 5$, $b = 20$. Результаты вычислений представим в виде таблицы (табл. 5). Здесь обозначения аналогичны используемым в блок-схеме (см. рис. 31). Расчеты проводятся в соответствии с блок-схемой с погрешностью $\varepsilon = 1$ км/ч.

Таблица 5

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

Приведем решение для первого этапа:

$$y = 0.618 \cdot 5 + 0.382 \cdot 20 \approx 10.7,$$

$$z = 0.382 \cdot 5 + 0.618 \cdot 20 \approx 14.3,$$

$$A = 24 - \frac{2}{3} \cdot 10.7 + \frac{1}{30} \cdot 10.7^2 \approx 20.7,$$

$$B = 24 - \frac{2}{3} \cdot 14.3 + \frac{1}{30} \cdot 14.3^2 \approx 21.3,$$

$$A < B.$$

При данной невысокой точности вычислений достаточно четырех шагов оптимизации. В этом случае искомое значение скорости равно $v = (8.6 + 10.7)/2 = 9.65$ км/ч. После

пяти шагов этот результат получается с меньшей погрешностью:

$$v = (9.4 + 10.7)/2 = 10.05 \text{ км/ч}.$$

§ 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

$$\frac{\partial f}{\partial x_1} = 0, \quad \frac{\partial f}{\partial x_2} = 0, \ldots, \frac{\partial f}{\partial x_n} = 0. \tag{6.10}$$

Пример. В § 1 (п. 3) была рассмотрена задача об определении оптимальных размеров контейнера объемом 1 м^3. Задача свелась к минимизации его полной поверхности, которая в данном случае является целевой функцией

$$S = 2\left(x_1 x_2 + \frac{1}{x_1} + \frac{1}{x_2}\right). \tag{6.11}$$

Решение. В соответствии с (6.10) 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 (6.10).

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.

Следует отметить, что такой метод может быть использован для функции одной переменной. 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 общего поиска 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 10^{15}, т. е. приблизительно 10^{10}. Число арифметических операций тогда равно 10^{12}, и для решения этой задачи на ЭВМ с быстродействием 1 млн. оп./с потребуется 10^6 с (более 10 сут) 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 координа-

тами $x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$. Зафиксируем все координаты функции u, кроме первой. Тогда $u = f(x_1, x_2^{(0)}, \ldots, x_n^{(0)})$ — функция одной переменной x_1. Решая одномерную задачу оптимизации для этой функции, мы от точки M_0 переходим к точке $M_1(x_1^{(1)}, x_2^{(0)}, \ldots, x_n^{(0)})$, в которой функция u принимает наименьшее значение по координате x_1 при фиксированных остальных координатах. В этом состоит первый шаг процесса оптимизации, состоящий в спуске по координате x_1.

Зафиксируем теперь все координаты, кроме x_2, и рассмотрим функцию этой переменной $u = f(x_1^{(1)}, x_2, x_3^{(0)}, \ldots, x_n^{(0)})$. Снова решая одномерную задачу оптимизации, находим ее наименьшее значение при $x_2 = x_2^{(1)}$, т. е. в точке $M_2(x_1^{(1)}, x_2^{(1)}, x_3^{(0)}, \ldots, x_n^{(0)})$. Аналогично проводится спуск по координатам $x_3, x_4, \ldots, x_n$, а затем процедура снова повторяется от x_1 до x_n и т. д. В результате этого процесса получается последовательность точек $M_0, M_1, \ldots$, в которых значения целевой функции составляют монотонно убывающую последовательность $f(M_0) \geqslant f(M_1) \geqslant \ldots$ На любом k-м шаге этот процесс можно прервать, и значение $f(M_k)$ принимается в качестве наименьшего значения целевой функции в рассматриваемой области.

Таким образом, метод покоординатного спуска сводит задачу о нахождении наименьшего значения функции многих переменных к многократному решению одномерных задач оптимизации по каждому проектному параметру.

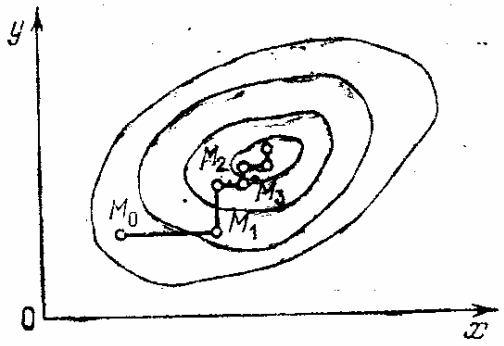


Рис. 32. Спуск по координатам

Данный метод легко проиллюстрировать геометрически для случая функции двух переменных $z = f(x, y)$, описывающей некоторую поверхность в трехмерном пространстве. На рис. 32 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.

Важным здесь является вопрос о сходимости рассматриваемого процесса оптимизации. Другими словами, будет ли последовательность значений целевой функции $f(M_0)$, $f(M_1)$, ... сходиться к наименьшему ее значению в данной области? Это зависит от вида самой функции и выбора начального приближения.

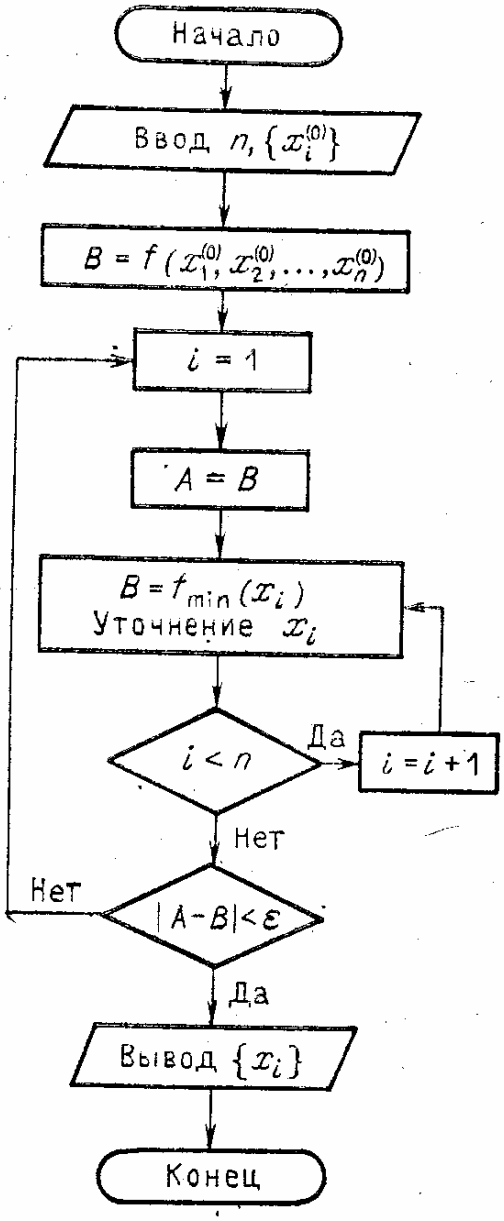


Рис. 33. Блок-схема метода покоординатного спуска

Для функции двух переменных очевидно, что метод неприменим в случае наличия изломов в линиях уровня. Это соответствует так называемому *оврагу* на поверхности. Здесь возможен случай, когда спуск по одной координате приводит на «дно» оврага. Тогда любое движение вдоль другой координаты ведет к возрастанию функции, соответствующему подъему на «берег» оврага. Поскольку поверхности типа «оврага» встречаются в инженерной практике, то при использовании метода покоординатного спуска следует убедиться, что решаемая задача не имеет этого недостатка.

Для гладких функций при удачно выбранном начальном приближении (в некоторой окрестности минимума) процесс сходится к минимуму. К достоинствам метода покоординатного спуска следует также отнести возможность использования простых алгоритмов одномерной оптимизации. Блок-схема метода покоординатного спуска представлена на рис. 33.

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 ситуа-

цию, 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 путь, ведущий вниз вдоль наиболее крутой линии. Методы, основанные на выборе пути оптимизации с помощью градиента, называются *градиентными.*

Идея метода градиентного спуска состоит в следующем. Выбираем некоторую начальную точку и вычисляем в ней градиент рассматриваемой функции. Делаем шаг в направлении, обратном градиентному. В результате приходим в точку, значение функции в которой обычно меньше первоначального. Если это условие не выполнено, т. е. значение функции не изменилось либо даже возросло, то нужно уменьшить шаг. В новой точке процедуру повторяем: вычисляем градиент и снова делаем шаг в обратном к нему направлении. Процесс продолжается до получения наименьшего значения целевой функции. Момент окончания поиска наступит тогда, когда движение из полученной точки с любым шагом приводит к возрастанию значения целевой функции. Строго говоря, если минимум функции достигается внутри рассматриваемой области, то в этой точке градиент равен нулю, что также может служить сигналом об окончании процесса оптимизации.

В описанном методе требуется вычислять на каждом шаге оптимизации градиент целевой функции $f(x_1, x_2, \ldots, x_n)$:

$$\operatorname{grad} f = \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right\}.$$

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:

$$\frac{\partial f}{\partial x_i} \approx \frac{1}{\Delta x_i}[f(x_1, \ldots, x_i + \Delta x_i, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)], \quad i = 1, 2, \ldots, n.$$

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, в этом направлении делают не один шаг, а двигаются до тех пор, пока целевая функция убывает, достигая таким образом минимума в некоторой точке. В этой 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.

Заметим, что метод наискорейшего спуска сводит многомерную задачу оптимизации к последовательности одномерных задач на каждом шаге оптимизации, как и в случае 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.

Советуем читателю для лучшего понимания метода наискорейшего спуска построить блок-схему, аналогичную представленной на рис. 33 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

$$F(x, \beta) = f(x) + \frac{1}{\beta}\varphi(x), \quad x = \{x_1, x_2, \ldots, x_n\}. \quad (6.12)$$

При этом дополнительную (*штрафную*) функцию $\varphi(x)$ выберем таким образом, чтобы при $\beta \to 0$ решение вспомогательной задачи стремилось к решению исходной или, по крайней мере, чтобы их минимумы совпадали: $\min F(x, \beta) \to m$ при $\beta \to 0$.

Штрафная функция $\varphi(x)$ должна учитывать ограничения, которые задаются при постановке задачи оптимизации. В частности, если имеются ограничения-неравенства вида $g_j(x_1, x_2, \ldots, x_n) \geqslant 0$ $(j = 1, 2, \ldots, J)$, 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 $f(x)$ и $F(x, \beta)$ имеют один и тот же минимум. Если хотя бы одно неравенство не выполнится, то вспомогательная целевая функция $F(x, \beta)$ получает бесконечно большие добавки, и ее значения далеки от минимума функции $f(x)$. 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:

$$\begin{aligned} &g_i(x) = 0, \quad i = 1, 2, \ldots, I; \\ &h_j(x) \geqslant 0, \quad j = 1, 2, \ldots, J; \quad x = \{x_1, x_2, \ldots, x_n\}. \end{aligned} \quad (6.13)$$

В этом случае в качестве вспомогательной целевой функ-

ции, для которой формулируется задача безусловной оптимизации во всем n-мерном пространстве, принимают функцию

$$F(x, \beta) = f(x) + \frac{1}{\beta}\left\{\sum_{i=1}^{I} g_i^2(x) + \sum_{j=1}^{J} h_j^2(x)[1 - \operatorname{sign} h_j(x)]\right., \tag{6.14}$$

$$\beta > 0.$$

Здесь взята такая штрафная функция, что при выполнении условий (6.13) она обращается в нуль. Если же эти условия нарушены (т. е. $g_i(x) \neq 0$, $h_j(x) < 0$ и $\operatorname{sign} h_j(x) = -1$), то штрафная функция положительна. Она увеличивает целевую функцию $f(x)$ тем больше, чем больше нарушаются условия (6.13).

При малых значениях параметра β вне области D функция $F(x, \beta)$ сильно возрастает. Поэтому ее минимум может быть либо внутри D, либо снаружи вблизи границ этой области. В первом случае минимумы функций $F(x, \beta)$ и $f(x)$ совпадают, поскольку дополнительные члены в (6.14) равны нулю. Если минимум функции $F(x, \beta)$ находится вне D, то минимум целевой функции $f(x)$ лежит на границе D. Можно при этом построить последовательность $\beta_k \to 0$ такую, что соответствующая последовательность минимумов функции $F(x, \beta)$ будет стремиться к минимуму функции $f(x)$.

Таким образом, задача оптимизации для целевой функции $f(x)$ с ограничениями (6.13) свелась к последовательности задач безусловной оптимизации для вспомогательной функции (6.14), решение которых может быть проведено с помощью методов спуска. При этом строится итерационный процесс при $\beta \to 0$.

Укрупненная блок-схема решения задачи математического программирования с использованием метода штрафных функций представлена на рис. 34. В качестве исходных данных вводятся начальное приближение искомого вектора $x^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$, начальное значение параметра β и некоторое малое число ε, характеризующее точность расчета. На каждом шаге итерационного процесса определяется оптимальное значение x^* вектора 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 заданной малой величиной.

В этом случае точка 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 $\varphi(x^*)=0$.

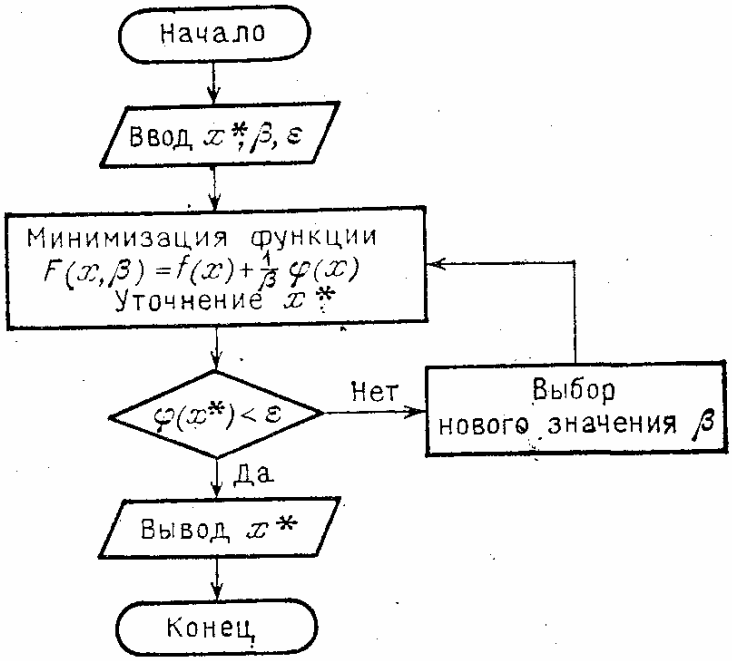


Рис. 34. Блок-схема метода штрафных функций

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 переменных $x_1, x_2, \ldots, x_n$, которые:

1) удовлетворяют системе линейных уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m; \end{aligned} \qquad (6.15)$$

2) являются неотрицательными, т. е.

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \ \ldots, \ x_n \geqslant 0; \tag{6.16}$$

3) обеспечивают наименьшее значение линейной целевой функции

$$f(x_1, x_2, \ldots, x_n) = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n. \tag{6.17}$$

Всякое решение системы уравнений (6.15), удовлетворяющее системе неравенств (6.16), называется *допустимым решением*. Допустимое решение, которое минимизирует целевую функцию (6.17), 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.

Введем 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	0.80	1.10	0.90
Вторая	1.00	0.70	1.20

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:

$$\begin{aligned} x_1 + x_2 + x_3 &= 12, \\ x_4 + x_5 + x_6 &= 15, \\ x_1 + x_4 &= 8, \\ x_2 + x_5 &= 9, \\ x_3 + x_6 &= 10, \end{aligned} \tag{6.18}$$

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

$$x_i \geqslant 0, \quad i = 1, 2, \ldots, 6, \tag{6.19}$$

которая означает, что товар обратно с магазинов на базы не вывозится. Общая стоимость перевозок с учетом приведенных в таблице расценок выразится формулой

$$f = 0.8x_1 + 1.1x_2 + 0.9x_3 + x_4 + 0.7x_5 + 1.2x_6. \tag{6.20}$$

Таким образом, мы пришли к типичной задаче линейного программирования: найти оптимальные значения проектных параметров x_i $(i = 1, 2, \ldots, 6)$, удовлетворяющих условиям (6.18), (6.19) и минимизирующих общую стоимость перевозок (6.20).

Из анализа системы уравнений (6.18) 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

$$\begin{aligned} x_1 + x_2 + x_3 &= 12, \\ x_4 + x_5 + x_6 &= 15, \\ x_1 + x_4 &= 8, \\ x_2 + x_5 &= 9. \end{aligned} \tag{6.21}$$

Число неизвестных на два больше числа уравнений, поэтому выразим через x_1 и x_2 все остальные неизвестные. Получим

$$\begin{aligned} x_3 &= 12 - x_1 - x_2, \\ x_4 &= 8 - x_1, \\ x_5 &= 9 - x_2, \\ x_6 &= x_1 + x_2 - 2. \end{aligned} \tag{6.22}$$

Поскольку в соответствии с (6.19) все проектные параметры должны быть неотрицательны, то с учетом (6.22)

получим следующую систему неравенств:

$$\begin{gathered} x_1 \geqslant 0, \quad x_2 \geqslant 0, \\ 12 - x_1 - x_2 \geqslant 0, \\ 8 - x_1 \geqslant 0, \quad 9 - x_2 \geqslant 0, \\ x_1 + x_2 - 2 \geqslant 0. \end{gathered} \tag{6.23}$$

Эти неравенства можно записать в более компактном виде:

$$0 \leqslant x_1 \leqslant 8, \quad 0 \leqslant x_2 \leqslant 9, \quad 2 \leqslant x_1 + x_2 \leqslant 12. \tag{6.24}$$

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 (6.20) для нее с учетом соотношений (6.22) принимает вид

$$f = 22.7 + 0.1x_1 + 0.7x_2. \tag{6.25}$$

Отсюда следует, что стоимость перевозок растет с увеличением значений x_1, x_2; поэтому нужно взять их наименьшие допустимые значения. В соответствии с (6.24)

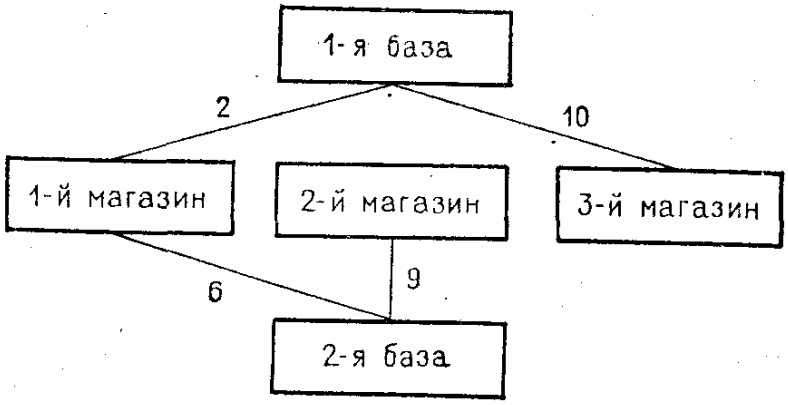


Рис. 35. Схема перевозок

$x_1 + x_2 \geqslant 2$; примем $x_1 + x_2 = 2$. Исключая один из параметров, например x_2, получим $x_2 = 2 - x_1$. Тогда

$$f = 24.1 - 0.6x_1.$$

Очевидно, что стоимость перевозок f будет минимальной, если величина x_1 примет наибольшее значение в рамках сделанного ограничения ($x_1 + x_2 = 2$). Таким оптимальным будет значение $x_1 = 2$. Тогда $x_2 = 0$, а опти-

мальные значения остальных проектных параметров можно найти по формулам (6.22): $x_3 = 10$, $x_4 = 6$, $x_5 = 9$, $x_6 = 0$. В этом случае минимальная общая стоимость перевозок f равна 22.9 р. На рис. 35 показана схема доставки товаров, соответствующая полученному решению. Числа указывают количество товара (в тоннах).

3. Геометрический метод. Областью решения линейного неравенства с двумя переменными

$$a_0 + a_1x_1 + a_2x_2 \geqslant 0 \qquad (6.26)$$

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 (6.26) имеет вид $a_0 + a_1x_1 \geqslant 0$; в этом случае получим либо $x_1 \geqslant h$ — правую полуплоскость, либо $x_1 \leqslant h$ — левую полуплоскость.

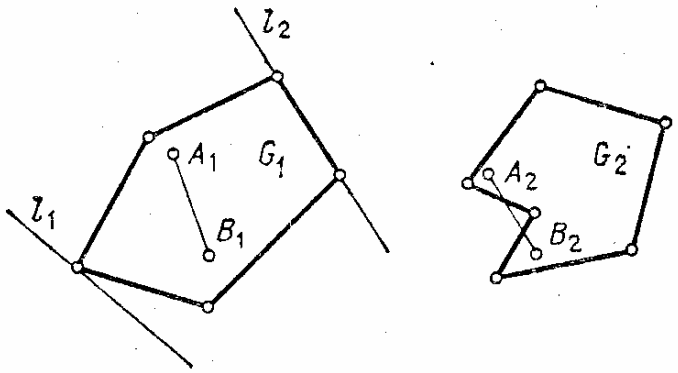


Рис. 36. Выпуклая (G_1) и невыпуклая (G_2) области

Областью решений системы неравенств является пересечение конечного числа полуплоскостей, описываемых каждым отдельным неравенством. 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. 36 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, все точки которого принадлежат обла-

сти 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. 36 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: $f = c_0 + c_1x_1 + c_2x_2 = C$. Это значение достигается в точках прямой, удовлетворяющих уравнению

$$c_0 + c_1x_1 + c_2x_2 = C. \qquad (6.27)$$

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.

Предположим, что прямая, записанная в виде (6.27), 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 (6.27) осуществляется в направлении, противоположном **n**, пока она не станет опорной. Тогда вершина многоугольника G, через которую проходит опорная прямая, будет соответствовать максимуму функции f. При дальнейшем переносе прямой целевая функция будет убывать.

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 заключение вернемся к рассмотренной ранее транспортной задаче (см. п. 2). На рис. 37 изображен многоугольник $ABCDEF$ допустимых решений. Он получен как пересечение полуплоскостей, описываемых неравенствами (6.23). Опорная прямая l_1 соответствует уравнению (6.25) при $f = 22.9$. Точка A пересечения опорной прямой с многоугольником решений дает минимум целевой функции.

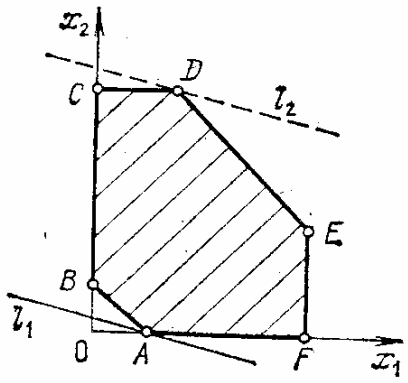


Рис. 37. Область допустимых решений

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 соответствую-

щей 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 ЭВМ.

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

$$f = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n \qquad (6.28)$$

при заданных ограничениях в виде системы линейных уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m. \end{aligned} \tag{6.29}$$

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 $x_{n+1} > 0$ 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 (6.29) *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 (6.29) 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 (6.29):

$$\begin{aligned} x_1 &= p_1 + q_{1,\,m+1}x_{m+1} + q_{1,\,m+2}x_{m+2} + \ldots + q_{1n}x_n, \\ x_2 &= p_2 + q_{2,\,m+1}x_{m+1} + q_{2,\,m+2}x_{m+2} + \ldots + q_{2n}x_n, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_m &= p_m + q_{m,\,m+1}x_{m+1} + q_{m,\,m+2}x_{m+2} + \ldots + q_{mn}x_n, \\ p_i &\geqslant 0, \quad i = 1, 2, \ldots, m. \end{aligned} \tag{6.30}$$

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 (6.30), можно выразить линейную целевую функцию (6.28) через свободные пере-

менные:

$$f = d_0 + d_{m+1}x_{m+1} + \ldots + d_n x_n. \qquad (6.31)$$

Процесс оптимизации начнем с некоторого начального (*опорного*) решения, например при нулевых значениях свободных переменных. Тогда получим

$$x_1 = p_1, \ \ldots, \ x_m = p_m, \ x_{m+1} = 0, \ \ldots, \ x_n = 0. \qquad (6.32)$$

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 (6.32) 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 (6.31) неотрицательны, то при увеличении любой свободной переменной $x_{m+1}, \ \ldots, \ x_n$ целевая функция не может уменьшиться. В этом случае решение (6.32) окажется оптимальным.

Пусть теперь среди коэффициентов формулы (6.31) хотя бы один отрицательный, например $d_{m+1} < 0$. Это означает, что при увеличении переменной x_{m+1} целевая функция уменьшается по сравнению со значением d_0, соответствующим решению (6.32). Поэтому в качестве нового опорного выбирается решение при следующих значениях свободных параметров:

$$x_{m+1} = x_{m+1}^{(1)}, \ x_{m+2} = 0, \ \ldots, \ x_n = 0, \qquad (6.33)$$

При этом базисные переменные, вычисляемые по формулам (6.30), равны

$$x_i = p_i + q_{i,m+1}x_{m+1}^{(1)}, \quad i = 1, 2, \ \ldots, \ m. \qquad (6.34)$$

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 отрицатель-

ные, а это влечет за собой угрозу сделать некоторые переменные x_i в (6.34) 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

$$p_i + q_{i,m+1}x_{m+1}^{(1)} \geqslant 0, \quad i = 1, 2, \ldots, m. \tag{6.35}$$

Среди всех отрицательных коэффициентов $q_{i,m+1}$ найдем наибольший по модулю. Пусть его значение равно Q, а соответствующее ему значение p_i равно P. Тогда из (6.35) 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

$$\begin{gathered} x_1 = p_1 - \frac{P}{Q} q_{1,m+1}, \ \ldots, \quad x_m = p_m - \frac{P}{Q} q_{m,m+1}, \\ x_{m+1} = -\frac{P}{Q}, \quad x_{m+2} = 0, \ \ldots, \ x_n = 0. \end{gathered} \tag{6.36}$$

Новая целевая функция при этих значениях проектных параметров равна

$$f^{(1)} = d_0 - d_{m+1}\frac{P}{Q}. \tag{6.37}$$

Полученное значение целевой функции $f^{(1)}$ меньше предыдущего, поскольку в данной формуле второй член правой части больше нуля $(d_{m+1} < 0,\ Q < 0,\ P > 0)$.

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 $x_1, \ldots, x_{m-1}, x_{m+1}$. После второго шага мы либо найдем новые оптимальные значения переменных и соответствующее им значение целевой функции $f^{(2)} < f^{(1)}$, либо покажем, что решение (6.36) 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 10 р., для его изготовления необходимо 4 кг металла, 2 м^2 стекла и 2 чел.-ч рабочего времени. Цена одного изделия Б 12 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:

$$\begin{aligned} 4x_1 + 5x_2 &\leqslant 300, \\ 2x_1 + x_2 &\leqslant 100, \\ 2x_1 + 3x_2 &\leqslant 160. \end{aligned} \tag{6.38}$$

Полная стоимость запланированной к производству продукции выражается формулой

$$f = 10x_1 + 12x_2. \tag{6.39}$$

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 (6.38) и дающих максимальное значение линейной целевой функции (6.39).

Вид сформулированной задачи не является каноническим, поскольку условия (6.38) 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 (6.38). При этом выбирают эти переменные такими, чтобы при их прибавлении к левым частям соотношений (6.38) неравенства превращались в равенства. Тогда ограничения примут вид

$$\begin{aligned} 4x_1 + 5x_2 + x_3 &= 300, \\ 2x_1 + x_2 + x_4 &= 100, \\ 2x_1 + 3x_2 + x_5 &= 160, \end{aligned} \tag{6.40}$$

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 (6.39), 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 (6.39) взять со знаком минус, т. е. принять целевую функцию в виде

$$F = -10x_1 - 12x_2, \tag{6.41}$$

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 (6.40). Получим

$$\begin{aligned} x_3 &= 300 - 4x_1 - 5x_2, \\ x_4 &= 100 - 2x_1 - x_2, \\ x_5 &= 160 - 2x_1 - 3x_2. \end{aligned} \tag{6.42}$$

В качестве опорного решения возьмем такое, которое соответствует нулевым значениям свободных параметров:

$$x_1^{(0)} = 0, \quad x_2^{(0)} = 0, \quad x_3^{(0)} = 300, \quad x_4^{(0)} = 100, \quad x_5^{(0)} = 160. \tag{6.43}$$

Этому решению соответствует нулевое значение целевой функции (6.41):

$$F^{(0)} = 0. \tag{6.44}$$

Исследуя полученное решение, отмечаем, что оно не является оптимальным, поскольку значение целевой функции (6.41) может быть уменьшено по сравнению с (6.44) путем увеличения свободных параметров.

Положим $x_2 = 0$ и будем увеличивать переменную x_1 до тех пор, пока базисные переменные остаются положительными. Из (6.42) следует, что x_1 можно увеличить до значения $x_1 = 50$, поскольку при большем его значении переменная x_4 станет отрицательной.

Таким образом, полагая $x_1 = 50$, $x_2 = 0$, получаем новое опорное решение (значения переменных x_3, x_4, x_5 найдем по формулам (6.42)):

$$x_1^{(1)} = 50, \quad x_2^{(1)} = 0, \quad x_3^{(1)} = 100, \quad x_4^{(1)} = 0, \quad x_5^{(1)} = 60. \tag{6.45}$$

Значение целевой функции (6.41) при этом будет равно

$$F^{(1)} = -500. \tag{6.46}$$

Новое решение (6.45), следовательно, лучше, поскольку значение целевой функции уменьшилось по сравнению с (6.44).

Следующий шаг начнем с выбора нового базиса. Примем ненулевые переменные в (6.45) x_1, x_3, x_5 в качестве базисных, а нулевые переменные x_2, x_4 в качестве свободных. Из системы (6.40) найдем

$$\begin{aligned} x_1 &= 50 - \frac{1}{2}x_2 - \frac{1}{2}x_4, \\ x_3 &= 100 - 3x_2 + 2x_4, \\ x_5 &= 60 - 2x_2 + x_4. \end{aligned} \tag{6.47}$$

Выражение для целевой функции (6.41) запишем через свободные параметры, заменив x_1 с помощью (6.47). Получим

$$F = -500 - 7x_2 + 5x_4. \tag{6.48}$$

Отсюда следует, что значение целевой функции по сравнению с (6.46) можно уменьшить за счет увеличения x_2, поскольку коэффициент при этой переменной в (6.48) 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 (6.47). 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 (6.47):

$$x_1^{(2)} = 35, \quad x_2^{(2)} = 30, \quad x_3^{(2)} = 10, \quad x_4^{(2)} = 0, \quad x_5^{(2)} = 0. \tag{6.49}$$

При этом значение целевой функции (6.48) равно

$$F^{(2)} = -710. \tag{6.50}$$

Покажем, что полученное решение является оптимальным. Для проведения следующего шага ненулевые переменные в (6.49), 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

$$F = -710 + \frac{3}{2}x_4 + \frac{7}{2}x_5.$$

Поскольку коэффициенты при x_4, x_5 положительные, то при увеличении этих параметров целевая функция возрастает. Следовательно, минимальное значение целевой функции $F_{\min} = -710$ 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 710 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

$$Q = \rho v \left(1 - v^2/v_{\max}^2\right)^{1/(k-1)}.$$

При какой скорости v расход газа будет максимальным?

4. Составить блок-схему 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. Спроектировать цилиндрический котел емкостью 200 л таким образом, чтобы на его изготовление было израсходовано как можно меньше материала.

9. Начертить области, определенные системами неравенств:

а) $x \geqslant 0,\ y \geqslant 0,\ 2x + y \leqslant 4$;

б) $x - y \geqslant 0,\ x \leqslant 9,\ x + 3y \geqslant 6$.

10. Минимизировать функцию $f = 12x_1 + 4x_2$ при наличии ограничений $x_1 + x_2 \geqslant 2,\ x_1 > 0.5,\ x_2 \leqslant 4,\ x_1 - x_2 \geqslant 0$.

11. 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 70 к., с первого склада на второй завод — 90 к., со второго склада на первый завод — 1 р., со второго склада на второй завод — 80 к.

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 функции $y = y(x)$. Их можно записать в виде

$$F(x, y, y', \ldots, y^{(n)}) = 0, \qquad (7.1)$$

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,

$$y' = f(x, y), \qquad (7.2)$$
$$y'' = f(x, y, y').$$

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 функция $y = \varphi(x)$, которая после ее подстановки в уравнение превращает его в тождество.

Общее решение обыкновенного дифференциального уравнения n-го порядка содержит n произвольных постоянных $C_1, C_2, \ldots, C_n$, т. е. общее решение уравнения (7.1) имеет вид

$$y = \varphi(x, C_1, C_2, \ldots, C_n). \qquad (7.3)$$

Частное решение дифференциального уравнения получается из общего, если произвольным постоянным придать определенные значения.

Для уравнения первого порядка общее решение зависит от одной произвольной постоянной:

$$y = \varphi(x, C). \qquad (7.4)$$

Если постоянная принимает определенное значение $C = C_0$, то получим частное решение

$$y = \varphi(x, C_0).$$

Дадим геометрическую интерпретацию дифференциального уравнения первого порядка (7.2). Поскольку производная y' характеризует наклон касательной к интегральной кривой в данной точке, то при $y' = k = \text{const}$ из (7.1) 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. Через каждую точку из области решения проходит 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*.

Если же 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 области решения дифференциального уравнения.

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 (7.4). 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, однородными, линейными 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 быстродействующих ЭВМ, обладающих достаточно большим объемом оперативной памяти.

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

$$Y(0) = A, \quad Y(a) = 0, \quad Y(b) = 1, \quad Y(0) = A, \quad Y'(0) = B$$

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. $x = 0$, $x = a$, $x = b$ 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 которой для простоты будем считать постоянным. Совокупность узлов $x_0, x_1, \ldots$ 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 $g_h = \{x_i\}$ $(i = 0, 1, \ldots)$. 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 этой аппроксимации 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,

$$\varepsilon_h = \max_i |\varepsilon(x_i)|, \quad \varepsilon_h = \left[\sum_i \varepsilon^2(x_i)\right]^{1/2}.$$

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

$$l_h y_h = \varphi_h, \qquad x \in \gamma_h. \tag{7.8}$$

Здесь γ_h — граничные узлы сетки, т. е. $\gamma_h \in \Gamma$. 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

$$LY = dY/dx = F(x), \quad x > x_0, \quad Y(x_0) = A.$$

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 в виде

$$L_h y_h = \frac{y_{i+1} - y_i}{h} = f_i, \quad i = 0, 1, \ldots; \; y_0 = A.$$

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 ЭВМ 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 сеточной функции и искомой функции в узлах сетки, 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

$$L_h Y_h + L_h \delta_h = f_h, \quad x \in g_h,$$
$$l_h Y_h + l_h \delta_h = \varphi_h, \quad x \in \gamma_h.$$

Отсюда

$$L_h \delta_h = R_h, \quad l_h \delta_h = r_h.$$

Здесь $R_h = f_h - L_h Y_h$ — погрешность аппроксимации (*невязка*) для разностного уравнения, а $r_h = \varphi_h - l_h Y_h$ — 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 $r=O(h^k)$ 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 $\delta \leqslant Mh^k$, то говорят, что разностная схема имеет точность *k-го порядка* или *сходится со скоростью* $O(h^k)$. Здесь $M>0$ — некоторая постоянная величина, не зависящая от h. Предполагается также, что $h>0$; в противном случае в указанных оценках необходимо взять $|h|$.

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

$$dY/dx = f(x,\ Y) \tag{7.9}$$

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 $x>x_0$.

Из курса дифференциальных уравнений известно, что 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

$$Z'' = \varphi(Z', Z, x) \tag{7.11}$$

можно записать в виде системы уравнений

$$Y_1' = \varphi(Y_1, Y_2, x), \quad Y_2' = Y_1, \tag{7.12}$$

где $Y_1 = Z'$, $Y_2 = Z$.

Систему (7.12) можно записать с помощью одного векторного уравнения

$$Y' = f(Y, x). \tag{7.13}$$

Здесь

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} Z' \\ Z \end{bmatrix}, \quad f = \begin{bmatrix} \varphi \\ Z \end{bmatrix}.$$

Таким образом, векторное уравнение (7.13) может заменить как систему уравнений, так и уравнение 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 $\{x_i, y_i\}$ $(i = 0, 1, \ldots)$, называют *сеточной функцией*.

Далее, заменяя 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:

$$y_{i+1} = F(x_i, h_i, y_{i+1}, y_i, \ldots, y_{i-k+1}), \quad i = 1, 2, \ldots, \tag{7.14}$$

$$y_0 = Y_0. \tag{7.15}$$

Здесь разностное уравнение (7.14) записано в общем виде, а конкретное выражение его правой части зависит от способа аппроксимации производной. Для каждого численного метода получается свой вид уравнения (7.14).

На основании анализа вида разностного уравнения можно провести некоторую классификацию численных методов решения задачи Коши для обыкновенных дифференциальных уравнений.

Если в правой части (7.14) отсутствует y_{i+1}, т. е. значение y_{i+1} явно вычисляется по k предыдущим значениям y_i, y_{i-1}, ..., 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 (7.14) входит искомое значение y_{i+1}, то решение этого уравнения усложняется. В таких методах, называемых *неявными*, приходится решать уравнение (7.14) относительно y_{i+1} с помощью итерационных методов.

2. Одношаговые методы. Простейшим численным методом решения задачи Коши для обыкновенного дифференциального уравнения является *метод Эйлера*. Он основан на разложении искомой функции $Y(x)$ в ряд Тейлора в окрестностях узлов $x=x_i$ $(i=0, 1, \ldots)$, в котором отбрасываются все члены, содержащие производные второго и более высоких порядков. Запишем это разложение в виде

$$Y(x_i+\Delta x_i)=Y(x_i)+Y'(x_i)\,\Delta x_i+O(\Delta x_i^2). \quad (7.16)$$

Заменяем значения функции Y в узлах x_i значениями сеточной функции y_i. Кроме того, используя уравнение (7.9), полагаем

$$Y'(x_i)=f(x_i,\ Y(x_i))=f(x_i,\ y_i).$$

Будем считать для простоты узлы равноотстоящими, т. е. $\Delta x_i=x_{i+1}-x_i=h=\text{const}$ $(i=0, 1, \ldots)$. Учитывая введенные обозначения и пренебрегая членами порядка $O(h^2)$, из равенства (7.16) получаем

$$y_{i+1}=y_i+hf(x_i, y_i), \quad i=0, 1, \ldots \quad (7.17)$$

Полагая $i=0$, с помощью соотношения (7.17) 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. $y_0=Y(x_0)=Y_0$. Аналогично могут быть найдены значения сеточной функции в других узлах:

$$y_2=y_1+hf(x_1, y_1),$$
$$\ldots\ldots\ldots$$
$$y_n=y_{n-1}+hf(x_{n-1}, y_{n-1}).$$

Построенный алгоритм называется методом Эйлера. Разностная схема этого метода представлена соотношениями (7.17). 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.

Блок-схема алгоритма решения задачи Коши (7.9), (7.10) методом Эйлера изображена на рис. 38. Задаются начальные значения x, y_0, а также величина шага h и количество расчетных точек n. Решение получается в узлах $x+h$, $x+2h$, ..., $x+nh$. Вывод результатов предусмотрен на каждом шаге. Если найденные значения необходимо хранить в памяти машины, то следует ввести массив значений y_0, y_1, ..., y_n.

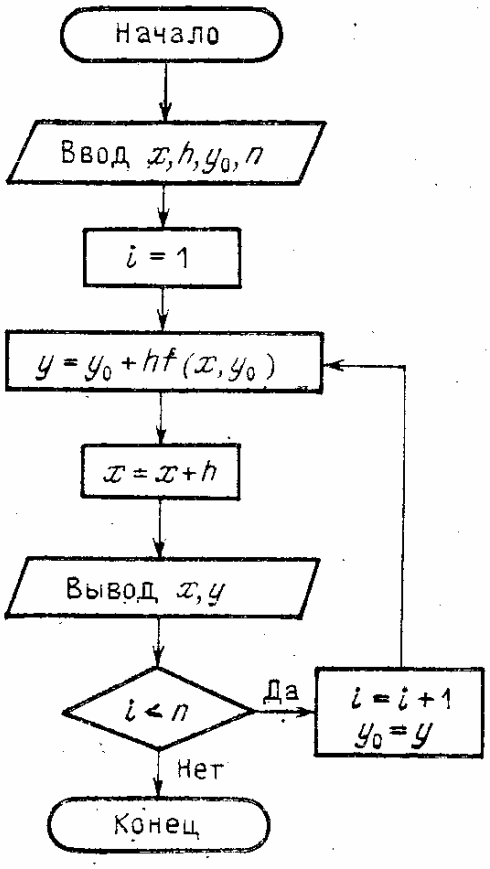


Рис. 38. Блок-схема метода Эйлера

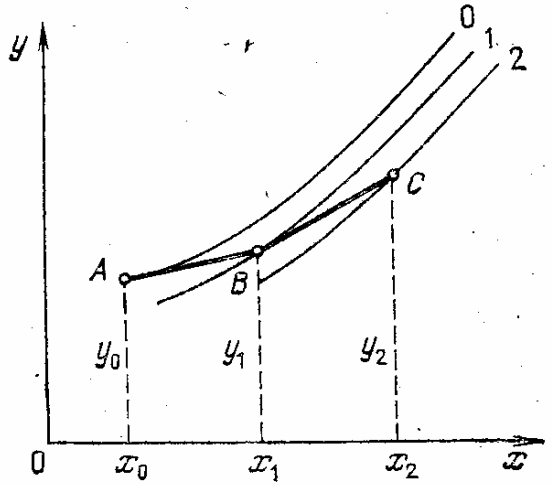


Рис. 39. Иллюстрация метода Эйлера

На рис. 39 дана геометрическая интерпретация метода Эйлера. 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 за-

дачи 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 $y_0' = f(x_0, y_0)$. Касательная BC уже проводится к другой интегральной кривой 1. Таким образом, погрешность метода Эйлера приводит к тому, что на каждом шаге решение переходит на другую интегральную кривую. Рассмотрим подробнее вопрос о погрешности метода Эйлера.

Погрешность e_i в точке x_i равна разности между значением сеточной функции y_i и точным значением искомой функции $Y(x_i)$: $e_i = y_i - Y(x_i)$. Эта погрешность состоит из двух частей: $e_i = e_i' + e_i''$. Составляющая e_i' определяется погрешностью начального значения $e_0 = y_0 - Y(x_0)$. Как правило, начальное значение задается точно, т. е. $y_0 = Y(x_0)$, и тогда $e_0 = 0$ и следовательно, равна нулю та часть погрешности решения e_i', которая связана с e_0. Погрешность e_i'' обусловлена отброшенными членами в разложении в ряд Тейлора (7.16). На каждом шаге эта погрешность имеет порядок $O(h^2)$, так как именно члены такого порядка отброшены в (7.16).

При нахождении решения в точке x_n, отстоящей на конечном расстоянии L от точки x_0, погрешность, в чем легко убедиться, суммируется. Суммарная погрешность, очевидно, равна $nO(h^2)$. Если учесть, что $h = L/n$, то для суммарной погрешности получаем окончательное выражение:

$$nO(h^2) = \frac{L}{h} O(h^2) = O(h). \qquad (7.18)$$

Таким образом, мы показали, что метод Эйлера имеет первый порядок точности.

Дадим еще одну схему метода Эйлера. Значение правой части $f(x, Y)$ уравнения (7.9) в схеме (7.17) возьмем равным среднему арифметическому значению между $f(x_i, y_i)$ и $f(x_{i+1}, y_{i+1})$, т. е. вместо разностной схемы (7.17) напишем

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1})], \quad i = 0, 1, \ldots \qquad (7.19)$$

Полученная схема является неявной, поскольку искомое значение y_{i+1} входит в обе части соотношения (7.19)

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 (7.17):

$$\tilde{y}_{i+1} = y_i + hf(x_i, y_i). \qquad (7.20)$$

Новое значение $\tilde{y}_{i+1}$ подставляем вместо y_{i+1} в правую часть соотношения (7.19) и находим окончательное значение

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, \tilde{y}_{i+1})]. \qquad (7.21)$$

Алгоритм (7.20), (7.21) можно записать в виде одного соотношения:

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))], \qquad (7.22)$$

$$i = 0, 1, \ldots$$

Эти рекуррентные соотношения описывают новую разностную схему, являющуюся модификацией метода Эйлера, которая называется *методом Эйлера с пересчетом*. Можно показать, используя разложение в ряд Тейлора, что этот метод имеет второй порядок точности. Его применение к решению задачи Коши уменьшает в среднем значения погрешностей до величин $O(h^2)$ вместо $O(h)$ в обычном методе Эйлера.

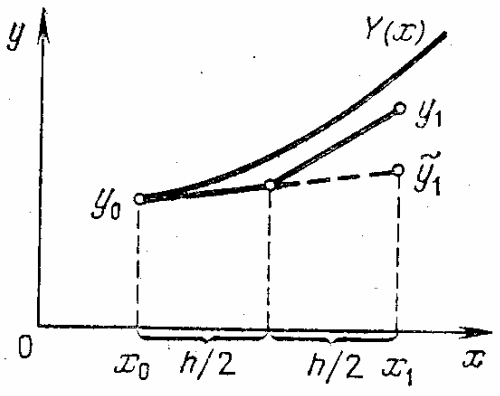


Рис. 40. Модифицированный метод Эйлера

На рис. 40 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 $y'_0 = f(x_0, y_0)$. С ее помощью методом Эйлера (7.17) 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 от-

резок касательной от точки $x_0 + h/2$ до точки x_1. В результате получается уточненное значение искомой функции y_1 в этой точке.

Метод Эйлера с пересчетом можно получить и иначе, используя разложение функции в ряд Тейлора. Запишем это разложение в виде

$$y_{i+1} = y_i + hy_i' + \frac{h^2}{2} y_i'' + O(h^3). \qquad (7.23)$$

В этой схеме должен быть сохранен член с h^2. Для этого аппроксимируем вторую производную с помощью отношения конечных разностей:

$$y_i'' = \frac{y_{i+1}' - y_i'}{h} + O(h).$$

Подставляя это соотношение в (7.23), получаем

$$y_{i+1} = y_i + \frac{h}{2}(y_i' + y_{i+1}') + O(h^3). \qquad (7.24)$$

Заменяя производные выражениями

$$y_i' = f(x_i, y_i),\ y_{i+1}' = f(x_{i+1}, \tilde{y}_{i+1}), \qquad (7.25)$$

где $\tilde{y}_{i+1}$ найдено по формуле (7.17) метода Эйлера, приходим к разностной схеме (7.21) метода Эйлера с пересчетом. Такой способ вывода формулы (7.21) позволил получить оценку погрешности метода. В соответствии с (7.24) погрешность на каждом шаге (локальная) имеет порядок h^3, а суммарная — порядок h^2.

Заметим, что при использовании неявной схемы (7.19) получается практически то же значение y_1, причем наклон отрезка, соединяющий точки (x_0, y_0) и (x_1, y_1), постоянный и равен наклону касательной к кривой в точке $x_0 + h/2$. Однако применение схемы (7.19), требующей построения итерационного процесса для вычисления значения y_1, привело бы к значительному возрастанию времени счета на каждом шаге.

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, $|y_{i+1} - y_{i+1}| > 0.01|y_{i+1}'|$), значение h следует уменьшить. Используя

эти оценки, можно построить алгоритм метода Эйлера с пересчетом с автоматическим выбором шага. Рекомендуем читателю составить такой алгоритм и построить соответствующую блок-схему.

Существуют и другие явные одношаговые методы. Наиболее распространенным из них является *метод Рунге — Кутта*. На его основе могут быть построены разностные схемы разного порядка точности. Приведем схему Рунге — Кутта четвертого порядка. Запишем алгоритм этого метода в виде

$$y_{i+1} = y_i + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3), \quad i = 0, 1, \ldots,$$

$$k_0 = f(x_i, y_i), \quad k_1 = f(x_i + h/2, \quad y_i + k_0/2), \quad (7.26)$$

$$k_2 = f(x_i + h/2, \quad y_i + k_1/2), \quad k_3 = f(x_i + h, \; y_i + k_2).$$

Таким образом, метод Рунге — Кутта требует на каждом шаге четырехкратного вычисления правой части уравнения $f(x, y)$.

Метод Эйлера (7.17) и его модифицированный вариант (7.22) также могут рассматриваться как методы Рунге — Кутта первого и второго порядков. Метод Рунге — Кутта (7.26) требует большего объема вычислений, однако это окупается повышенной точностью, что дает возможность проводить счет с большим шагом. Другими словами, для получения результатов с одинаковой точностью в методе Эйлера потребуется значительно меньший шаг, чем в методе Рунге — Кутта.

Проведем сравнительную оценку рассмотренных методов на простом примере, позволяющем получить также и точное решение.

Пример. Решить задачу Коши

$$dY/dx = 2(x^2 + Y), \quad Y(0) = 1, \quad 0 \leqslant x \leqslant 1, \quad h = 0.1.$$

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

$$Y = 1.5e^{2x} - x^2 - x - 0.5.$$

Проведем теперь решение данной задачи численно с помощью рассмотренных выше методов. Результаты вы-

числений приведены в табл. 6. Как видно из этой таблицы, самым точным является решение, полученное методом Рунге — Кутта. Анализ решения с использованием метода Эйлера позволяет проследить рость погрешности с возрастанием x_i. При $x_i = 1$ погрешность составляет почти 18%. Следовательно, при большом числе узлов метод Эйлера может привести к заметным погрешностям, и в таких случаях предпочтительнее пользоваться численными методами высших порядков точности.

Таблица 6

x_i	Метод Эйлера	Модифицированный метод Эйлера	Метод Рунге—Кутта	Точное решение
0.1	1.2000	1.2210	1.2221	1.2221
0.2	1.4420	1.4923	1.4977	1.4977
0.3	1.7384	1.8284	1.8432	1.8432
0.4	2.1041	2.2466	2.2783	2.2783
0.5	2.5569	2.7680	2.8274	2.8274
0.6	3.1183	3.4176	3.5201	3.5202
0.7	3.8139	4.2257	4.3927	4.3928
0.8	4.6747	5.2288	5.4894	5.4895
0.9	5.7376	6.4004	6.8643	6.8645
1.0	7.0472	8.0032	8.5834	8.5836

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.

Рассмотренные методы могут быть использованы также для решения систем дифференциальных уравнений. Покажем это для случая системы двух уравнений вида

$$dY/dx = \varphi(x, Y, Z),$$

$$dZ/dx = \psi(x, Y, Z).$$

Начальные условия зададим в виде

$$Y(x_0) = y_0, \quad Z(x_0) = z_0.$$

По аналогии с (7.26) запишем формулы Рунге — Кутта для системы двух уравнений:

$$y_{i+1} = y_i + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3),$$

$$z_{i+1} = z_i + \frac{h}{6}(l_0 + 2l_1 + 2l_2 + l_3), \quad i = 0, 1, \ldots,$$

$$k_0 = \varphi(x_i, y_i, z_i), \quad l_0 = \varphi(x_i, y_i, z_i),$$

$$k_1 = \varphi\left(x_i + \frac{h}{2}, y_i + \frac{k_0}{2}, z_i + \frac{l_0}{2}\right),$$

$$l_1 = \psi\left(x_i + \frac{h}{2}, y_i + \frac{k_0}{2}, z_i + \frac{l_0}{2}\right),$$

$$k_2 = \varphi\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}, z_i + \frac{l_1}{2}\right),$$

$$l_2 = \psi\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}, z_i + \frac{l_1}{2}\right),$$

$$k_3 = \varphi(x_i + h, y_i + k_2, z_i + l_2), \quad l_3 = \psi(x_i + h, y_i + k_2, z_i + l_2).$$

К решению систем уравнений сводятся также задачи Коши для уравнения высших порядков. Например, рассмотрим задачу Коши для уравнения второго порядка

$$d^2Y/dx^2 = f(x, Y, dY/dx),$$

$$Y(x_0) = y_0, \quad Y'(x_0) = z_0.$$

Введем вторую неизвестную функцию $Z(x) = Y'(x)$. Тогда сформулированная задача Коши заменяется следующей:

$$dZ/dx = f(x, Y, Z),$$

$$dY/dx = Z,$$

$$Y(x_0) = y_0, \quad Z(x_0) = z_0.$$

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 $t = 0$ 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.

3. Многошаговые методы. Другой путь построения разностных схем основан на том, что для вычисления зна-

чения y_{i+1} используются результаты не одного, а k предыдущих шагов, т. е. значения y_{i-k+1}, y_{i-k+2}, ..., y_i. В этом случае получается k-шаговый метод.

Многошаговые методы могут быть построены следующим образом. Запишем исходное уравнение (7.9) в виде

$$dY(x) = f(x, Y)\,dx. \tag{7.27}$$

Проинтегрируем обе части этого уравнения по x на отрезке $[x_i, x_{i+1}]$. Интеграл от левой части легко вычисляется:

$$\int_{x_i}^{x_{i+1}} dY(x) = Y(x_{i+1}) - Y(x_i) \approx y_{i+1} - y_i. \tag{7.28}$$

Для вычисления интеграла от правой части уравнения (7.27) строится сначала интерполяционный многочлен $P_{k-1}(x)$ 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

$$\int_{x_i}^{x_{i+1}} f(x, Y)\,dx \approx \int_{x_i}^{x_{i+1}} P_{k-1}(x)\,dx. \tag{7.29}$$

Приравнивая выражения, полученные в (7.28) и (7.29), можно получить формулу для определения неизвестного значения сеточной функции y_{i+1} в узле x_{i+1}:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} P_{k-1}(x)\,dx. \tag{7.30}$$

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. 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

$$\Delta f_i = f_i - f_{i-1},$$
$$\Delta^2 f_i = f_i - 2f_{i-1} + f_{i-2},$$
$$\Delta^3 f_i = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}.$$

Тогда разностная схема четвертого порядка метода Адамса запишется в виде

$$y_{i+1} = y_i + hf_i + \frac{h^2}{2}\Delta f_i + \frac{5h^3}{12}\Delta^2 f_i + \frac{3h^4}{8}\Delta^3 f_i. \quad (7.31)$$

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 (метод Рунге — Кутта — четырех). Но метод Адамса неудобен тем, что невозможно начать счет по одному лишь известному значению y_0. Расчет может быть начат лишь с узла x_3. Значения y_1, y_2, y_3, необходимые для вычисления y_3, 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: 1) 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; 2) используя неявный метод (*корректор*), в результате итераций находятся приближения $y_{i+1}^{(1)}, y_{i+1}^{(2)}, \ldots$

Один из вариантов метода прогноза и коррекции может быть получен на основе метода Адамса четвертого порядка. Приведем окончательный вид разностных соотношений: на этапе предиктора

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}); \quad (7.32)$$

на этапе корректора

$$y_{i+1} = y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}). \quad (7.33)$$

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 с 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. Блок-схема решения задачи Коши с помощью рассмотренного метода прогноза и коррекции представлена на рис. 41.

4. 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*. Он состоит в том, что проводятся повторные расчеты по одной разностной схеме с различными шагами. Уточненное решение в совпадаю-

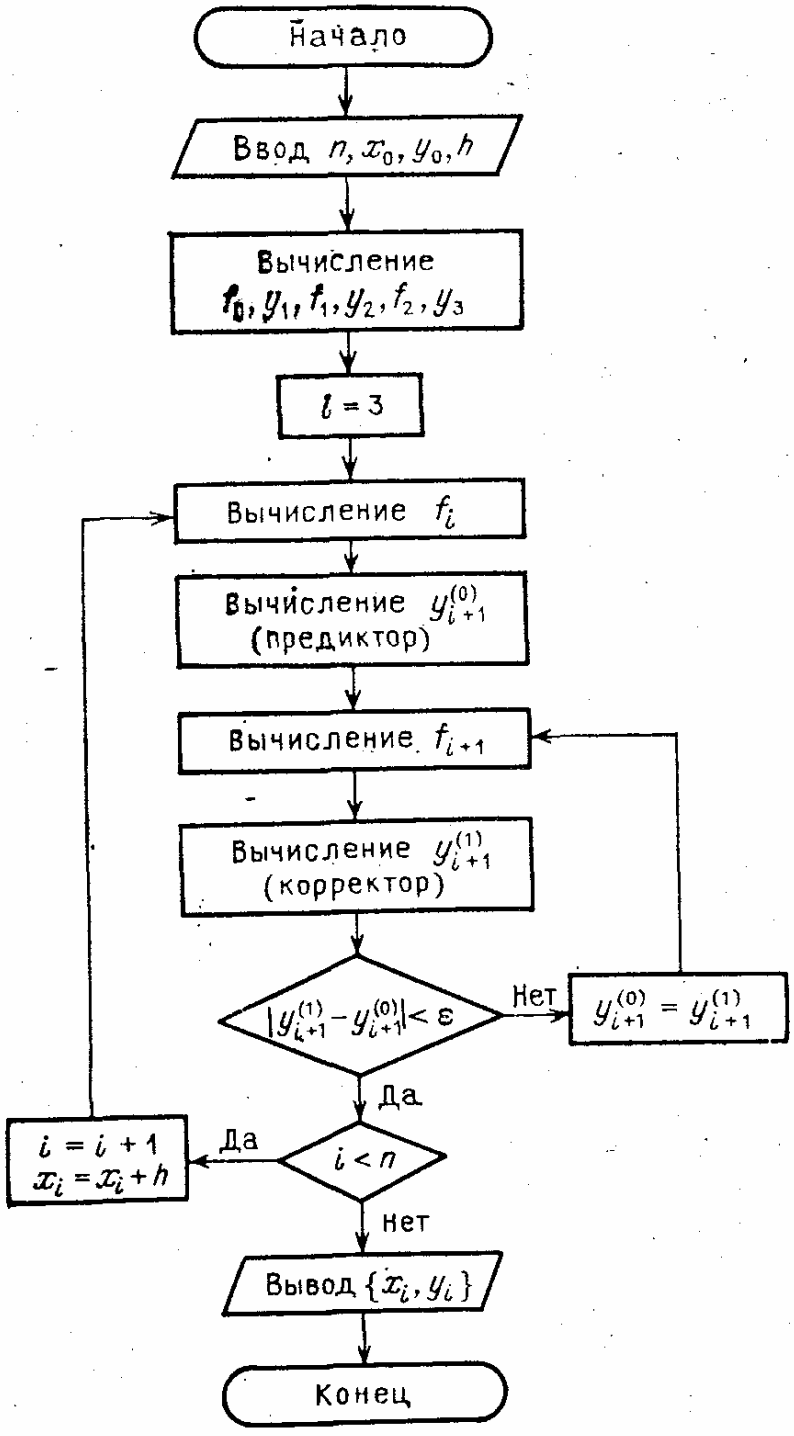


Рис. 41. Блок-схема метода предиктор-корректор

щих 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 $y_{h/2}$. Тогда в соответствии с методом Рунге уточненное значение y_h^* сеточной функции в узлах сетки

с шагом h вычисляется по формуле

$$y_h^* = \frac{2^k y_{h/2} - y_h}{2^k - 1} + O(h^{k+1}).$$

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

$$y_h^* = 2y_{h/2} - y_h + O(h^2).$$

Аналогично можно записать формулу для уточнения решения, полученного по методу Рунге — Кутта при $k=4$.

§ 3. Краевые задачи

1. Предварительные замечания. В § 2 рассматривались задачи с начальными условиями, т. е. с условиями в одной (начальной) точке: при $x=x_0$, $t=0$ и т. п. На практике приходится часто решать другого типа задачи, 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

$$Y'' + p(x)Y' + q(x)Y = f(x). \tag{7.34}$$

Краевая задача состоит в отыскании решения $Y=Y(x)$ уравнения (7.34) на отрезке $[a, b]$, удовлетворяющего на концах отрезка условиям

$$Y(a)=A, \quad Y(b)=B. \tag{7.35}$$

Граничные условия могут быть заданы не только в частном виде (7.35), но и в более общем виде:

$$\begin{aligned} \alpha_1 Y(a) + \beta_1 Y'(a) &= A, \\ \alpha_2 Y(b) + \beta_2 Y'(b) &= B. \end{aligned} \tag{7.36}$$

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 вычислительных машин. 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 (7.34) с граничными условиями (7.36) 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 (7.35), а $\varphi_1(x)$, ..., $\varphi_n(x)$ — соответствующим однородным граничным условиям. Искомое решение представляется в виде линейной комбинации базисных функций:

$$y(x) = \varphi_0(x) + a_1\varphi_1(x) + a_2\varphi_2(x) + \ldots + a_n\varphi_n(x). \quad (7.37)$$

Подставляя это выражение в уравнение (7.34), можно найти разность между его левой и правой частями, которая называется *невязкой*. Она является функцией переменной x и параметров $a_1, a_2, \ldots, a_n$ и имеет вид

$$\psi(x, a_1, a_2, \ldots, a_n) = Y'' + p(x)Y' + q(x) - f(x). \quad (7.38)$$

Коэффициенты $a_1, a_2, \ldots, a_n$ стараются подобрать так, чтобы невязка была 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 (7.38) в которых приравниваются нулю. Получается система n линейных алгебраических уравнений относительно $a_1, a_2, \ldots, a_n$. Решая данную систему, можно найти эти коэффициенты, которые затем подставляются в решение (7.37).

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 $\varphi_0(x)$, $\varphi_1(x), \ldots, \varphi_n(x)$ к невязке $\psi(x, a_1, \ldots, a_n)$, которое выражается в виде

$$\int_a^b \psi(x, a_1, \ldots, a_n)\varphi_i(x) = 0, \quad i = 1, 2, \ldots, n.$$

Из этих условий также получается система линейных алгебраических уравнений относительно коэффициентов линейного соотношения (7.37).

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.

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:

$$Y'' = f(x, Y, Y'). \tag{7.39}$$

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 (7.35), т. е.

$$Y(0) = y_0, \quad Y(1) = y_1. \tag{7.40}$$

Сущность *метода стрельбы* заключается в сведении решения краевой задачи (7.39), (7.40) к решению задач Коши для того же уравнения (7.39) с начальными условиями

$$Y(0) = y_0, \quad Y'(0) = k = \operatorname{tg}\alpha. \tag{7.41}$$

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. 42).

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 $(0, y_1)$. 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 (7.40), получаем $Y(1, \alpha) = y_1$, или

$$Y(1, \alpha) - y_1 = 0. \qquad (7.42)$$

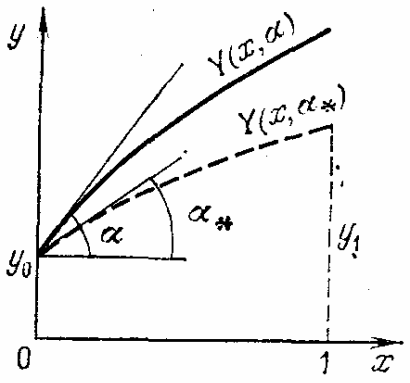


Рис. 42. Метод стрельбы

Следовательно, получим уравнение вида $F(\alpha) = 0$, где $F(\alpha) = Y(1, \alpha) - y_1$. Это уравнение отличается от привычной записи тем, что функцию $F(\alpha)$ нельзя представить в виде некоторого аналитического выражения, поскольку она является решением задачи Коши (7.39), (7.41). Тем не менее для решения уравнения (7.42) 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 $F(\alpha)$ принимает значения разных знаков. Для этого решение задачи Коши $Y(1, \alpha_0)$ должно при $x = 1$ находиться ниже точки y_1, а $Y(1, \alpha_1)$ 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 $F(\alpha)$ не меняет знак, и т. д. (см. блок-схему на рис. 24). Процесс поиска решения прекращается, если разность двух последовательно найденных значений α меньше некоторого наперед заданного малого числа. В этом случае последнее 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 на-

чальной 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.

Существуют другие алгоритмы метода стрельбы. В частности, одним из самых надежных является *метод Ньютона*. Он состоит в следующем. Пусть α_0 — начальное приближение α, а $\alpha_* = \alpha_0 + \Delta\alpha$ — искомое значение α. Решая задачу Коши при $\alpha = \alpha_0$, находим $Y(x, \alpha_0)$. Тогда можем записать разложение в ряд с сохранением только линейных по $\Delta\alpha$ членов:

$$Y(1, \alpha_0 + \Delta\alpha) \approx Y(1, \alpha_0) + \frac{\partial Y}{\partial \alpha} \Delta\alpha.$$

Полагая $Y(1, \alpha_0 + \Delta\alpha) = Y(1, \alpha_*) = y_1$, находим

$$\Delta\alpha = \frac{y_1 - Y(1, \alpha_0)}{\partial Y(1, \alpha_0)/\partial\alpha}. \tag{7.43}$$

Производную в знаменателе этого выражения можно найти численно:

$$\frac{\partial Y(1, \alpha_0)}{\partial \alpha} \approx \frac{Y(1, \alpha_0 + \delta\alpha) - Y(1, \alpha_0)}{\delta\alpha}. \tag{7.44}$$

Здесь $\delta\alpha$ — произвольное малое возмущение α.

Для вычисления правой части (7.44) нужно решить задачу Коши при $\alpha = \alpha_0 + \delta\alpha$, в результате чего найдем значение $Y(1, \alpha_0 + \delta\alpha)$. Вычисляя затем по формуле (7.43) поправку $\Delta\alpha$, находим следующее приближение параметра α: $\alpha_1 = \alpha_0 + \Delta\alpha$ и т. д. Этот итерационный процесс продолжается до тех пор, пока очередное значение поправки $\Delta\alpha$ по абсолютной величине не станет меньшим заданного малого числа ε.

Блок-схема решения краевой задачи методом стрельбы с применением пристрелки по методу Ньютона представлена на рис. 43. Решение задачи Коши 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 $x = 0, h, \ldots, \ldots, 1$, где $n = 1/h$.

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 $x=1$.

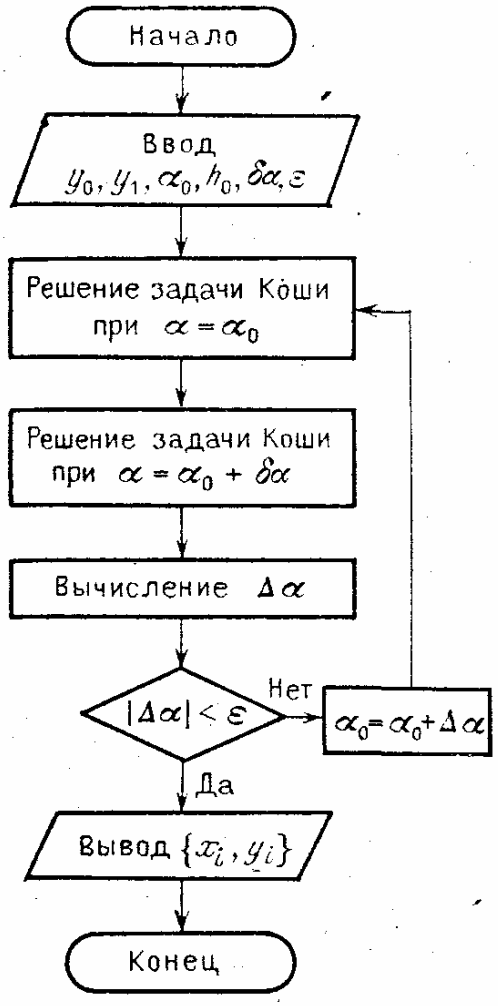


Рис. 43. Блок-схема метода стрельбы

Рассмотрим систему двух уравнений первого порядка

$$Y'=f_1(x, Y, Z),$$
$$Z'=f_2(x, Y, Z). \quad (7.45)$$

Граничные условия заданы в виде

$$Y(0)=y_0, \quad Z(1)=z_1. \quad (7.46)$$

Процесс решения этой краевой задачи методом стрельбы состоит в следующем. Выбирается некоторое α, аппроксимирующее значение $Z(0)$. Решается задача Коши для системы (7.45) 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 диф-

ференциальное уравнение, их конечно-разностными аппроксимациями (см. гл. 3, § 1).

Рассмотрим сущность такого метода решения для дифференциального уравнения второго порядка (7.39) при заданных граничных условиях (7.40). Разобьем отрезок $[0, 1]$ на n равных частей точками $x_i = ih$ $(i = 0, 1, \ldots, n)$. Решение краевой задачи (7.39), (7.40) сведем к вычислению значений сеточной функции y_i в узловых точках x_i. Для этого напишем уравнение (7.39) для внутренних узлов:

$$Y''(x_i) = f(x_i, Y(x_i), Y'(x_i)), \quad i = 1, 2, \ldots, n-1. \quad (7.47)$$

Заменим производные, входящие в эти соотношения, их конечно-разностными аппроксимациями:

$$Y'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h}, \quad Y''(x_i) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}. \quad (7.48)$$

Подставляя эти выражения в (7.47), получаем систему разностных уравнений

$$F(x_i, y_{i-1}, y_i, y_{i+1}) = 0, \quad i = 1, 2, \ldots, n-1, \quad (7.49)$$

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, если они задаются непосредственно.

На практике часто граничные условия задаются в более общем виде:

$$\begin{aligned} a_1 Y(0) + b_1 Y'(0) &= c_1, \\ a_2 Y(1) + b_2 Y'(1) &= c_2. \end{aligned} \quad (7.50)$$

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, при которых производные аппроксимируются с первым порядком точности, то разностные граничные условия примут вид

$$\begin{aligned} a_1 y_0 + b_1 \frac{y_1 - y_0}{h} &= c_1, \\ a_2 y_n + b_2 \frac{y_n - y_{n-1}}{h} &= c_2. \end{aligned} \quad (7.51)$$

Из этих соотношений легко находятся значения y_0, y_n.

Однако, как правило, предпочтительнее аппроксимировать производные, входящие в (7.50), со вторым порядком точности с помощью центральных разностей

$$Y'(0) = \frac{y_1 - y_{-1}}{2h}, \quad Y'(1) = \frac{y_{n+1} - y_{n-1}}{2h}, \qquad (7.52)$$

В эти выражения входят значения сеточной функции y_{-1} и y_{n+1} в так называемых *фиктивных узлах* $x_{-1} = -h$ и $x_{n+1} = 1 + h$, лежащих вне рассматриваемого отрезка. В этих узлах искомая функция также должна быть определена. Количество неизвестных значений сеточной функции при этом увеличивается на два. Для замыкания системы привлекают еще два разностных уравнения (7.49) — при $i = 0, n$.

Таким образом, решение краевой задачи для дифференциального уравнения сведено к решению системы алгебраических уравнений вида (7.49). Эта система является линейной или нелинейной в зависимости от того, линейно или нелинейно дифференциальное уравнение (7.39). 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

$$Y''(x) - p(x)Y(x) = f(x), \quad p(x) > 0, \quad 0 \leqslant x \leqslant 1, \qquad (7.53)$$

с граничными условиями вида

$$Y(0) = A, \quad Y(1) = B. \qquad (7.54)$$

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 (7.48). При этом значения искомой функции в узлах $Y(x_i)$ приближенно заменяем соответствующими значениями сеточной функции y_i. Записывая уравнение (7.53) 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

$$y_{i-1} - (2 + h^2 p_i) y_i + y_{i+1} = h^2 f_i, \quad i = 1, 2, \ldots, n-1. \tag{7.55}$$

Получилась система $n-1$ линейных уравнений, число которых совпадает с числом неизвестных значений сеточной функции $y_1, y_2, \ldots, y_{n-1}$ в узлах. Ее значения на концах отрезка определены граничными условиями (7.54):

$$y_0 = A, \quad y_n = B. \tag{7.56}$$

Решая систему уравнений (7.55) с учетом условий (7.56), 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 (7.55). 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. *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 (7.53), (7.54).

Система линейных алгебраических уравнений (7.55) 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:

$$Y'' = f(x, Y), \quad 0 \leqslant x \leqslant 1, \tag{7.57}$$

$$Y(0) = A, \quad Y(1) = B. \tag{7.58}$$

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

$$\partial f / \partial Y > 0. \tag{7.61}$$

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) в k-й итерации. Тогда, подставляя известные значения $y_i^{(k)}$ в правые части системы (7.59), получаем

$$y_{i-1}^{(k+1)} - 2y_i^{(k+1)} + y_{i+1}^{(k+1)} = h^2 f\left(x_i, y_i^{(k)}\right). \tag{7.62}$$

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 $y_i^{(0)}$ ($i = 1, 2, \ldots, n$); 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 (7.57).

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

$$d\Phi/dr = \mu I n h/(2\pi r).$$

Определить Ф при следующих данных: $I = 1$ А; $\mu = 80$; 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 блок-схему метода Эйлера (см. рис. 38) 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. Построить блок-схему решения задачи Коши для системы двух уравнений первого порядка методом Эйлера.

6. Составить блок-схему решения уравнения первого порядка методом Эйлера с пересчетом.

7. Построить алгоритм решения задачи Коши для уравнения второго порядка модифицированным методом Эйлера с автоматическим выбором шага.

8. С помощью итерационного метода предиктор-корректор найти решение при $x = 4h$ и $x = 5h$ ($h = 0.1$) для следующей задачи Коши:

$$dY/dt = t + \sin(Y/3), \quad Y(0) = 0{,}3.$$

9. Составить блок-схему решения краевой задачи методом стрельбы с использованием метода давления отрезка пополам.

10. Построить 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 ЭВМ большой мощности, т. е. обладающих большим объемом памяти и высокой скоростью вычислений.

Среди численных методов широко распространенными являются *разностные методы*. 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

$$\frac{\partial u}{\partial x} + p\frac{\partial u}{\partial y} = q, \qquad (8.2)$$

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)

$$\frac{\partial^2 u}{\partial t^2} = a^2\frac{\partial^2 u}{\partial x^2}; \qquad (8.3)$$

уравнение теплопроводности или *диффузии* (параболическое)

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2}, \quad a > 0; \qquad (8.4)$$

уравнение Лапласа (эллиптическое)

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0. \qquad (8.5)$$

Если правая часть уравнения (8.5) 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 Г в двумерном случае показан на рис. 44. 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 $y_i = c + jh_2$ $(j = 0, 1, \ldots, J)$. Через эти точки проводятся два семейства координатных прямых $x = \text{const}$ и $y = \text{const}$, образующих сетку с прямоугольной ячейкой. Любой узел этой сетки, номер которого (i, j), определяется координатами (x_i, y_j).

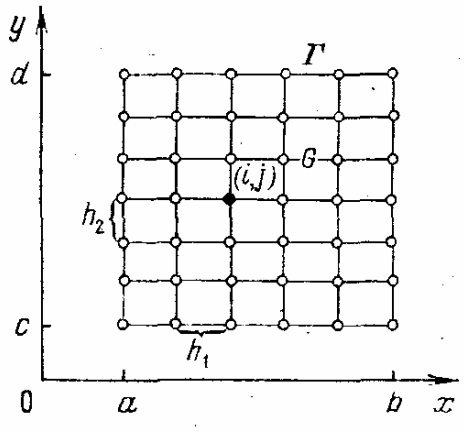


Рис. 44. Прямоугольная сетка

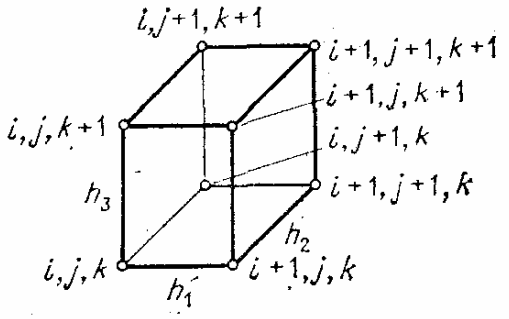


Рис. 45. Элемент сетки

Аналогично вводятся сетки для многомерных областей, содержащих более двух измерений. На рис. 45 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 треугольными и даже шестиугольными ячейками.

Узлы сетки, лежащие на границе Г 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.

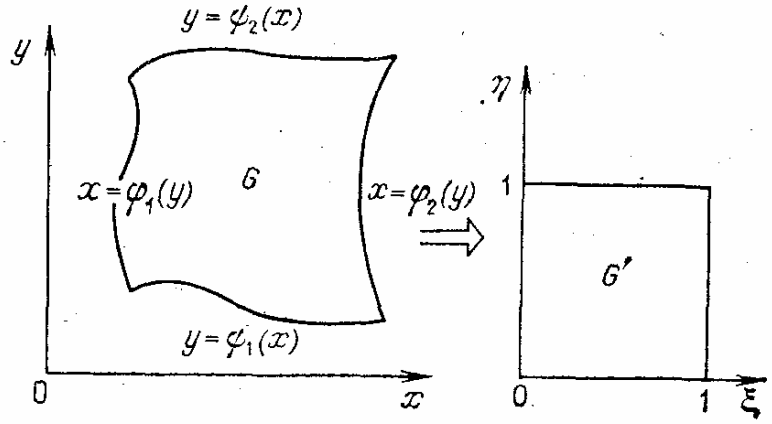


Рис. 46. Преобразование расчетной области

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. 46, 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).

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 U заменяются значениями сеточной функции u 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

$$\frac{\partial U}{\partial t} = a\frac{\partial^2 U}{\partial x^2}, \quad 0 \leqslant x \leqslant 1, \quad t > 0, \quad a > 0,$$

$$U(x, 0) = \varphi(x), \quad U(0, t) = \psi_1(t), \quad U(1, t) = \psi_2(t), \tag{8.6}$$

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 $x_i = ih$ $(i = 0, 1, \ldots, I)$, $t_j = j\tau$ $(j = 0, 1, \ldots, J)$; 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 соответ-

ствующими значениями сеточной функции u_i^j, которые удовлетворяют разностной схеме.

Заменяя в исходном уравнении (8.6) частные производные искомой функции с помощью отношений конечных разностей, получаем разностную схему

$$\frac{u_i^{j+1}-u_i^j}{\tau}=a\frac{u_{i+1}^j-2u_i^j+u_{i-1}^j}{h^2}, \tag{8.7}$$

$$i=1,\ 2,\ \ldots,\ I-1,\quad j=0,\ 1,\ \ldots.$$

В записи этой схемы для каждого узла использован шаблон, изображенный на рис. 47, *а*.

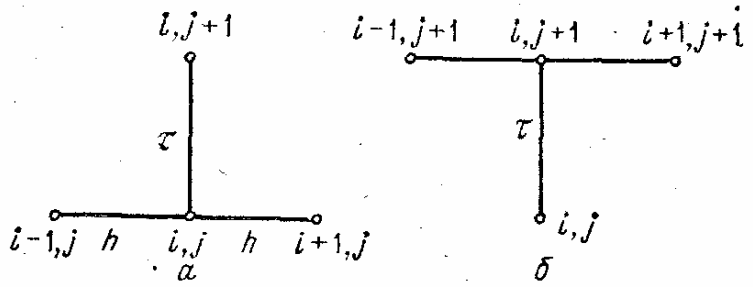


Рис. 47. Шаблоны

Для одного и того же уравнения можно построить различные разностные схемы. В частности, если воспользоваться шаблоном, изображенным на рис. 47, *б*, то вместо (8.7) получим разностную схему

$$\frac{u_i^{j+1}-u_i^j}{\tau}=a\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{h^2}. \tag{8.8}$$

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

$$u_0^j=\psi_1(t_j),\quad u_I^j=\psi_2(t_j). \tag{8.9}$$

Совокупность узлов при $t=\text{const}$, т. е. при фиксированном значении j, называется *слоем*. Схема (8.7) 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 при $j=1$ необходимо решение на начальном слое. Оно определяется начальным условием

(8.6), которое запишется в виде

$$u_i^0 = \varphi(x_i), \quad i = 1, 2, \ldots, I-1. \tag{8.10}$$

В отличие от явной схемы каждое разностное уравнение (8.8) содержит на каждом новом слое значения неизвестных в трех точках, поэтому нельзя сразу определить эти значения через известное решение на предыдущем слое. Такие схемы называются *неявными*. При этом разностная схема (8.8) состоит из линейных трехточечных уравнений, т. е. каждое уравнение содержит неизвестную функцию в трех точках данного слоя. Такие системы линейных уравнений с трехдиагональной матрицей могут быть решены методом прогонки (см. гл. 4, § 2), 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 понятия.

Исходную дифференциальную задачу, состоящую в решении уравнения с частными производными при заданных начальных и граничных условиях, запишем в операторном виде:

$$LU(x, t) = F(x, t), \quad (x, t) \in \bar{G}. \tag{8.11}$$

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 $\bar{G}$ включает расчетную область G и границу Г.

Дифференциальную задачу (8.11) заменяем разностной задачей относительно сеточной функции u, определенной в узлах сетки $\bar{g}_h$. Для простоты будем считать сетку зависящей от одного параметра h. Шаг по времени τ выражается через h: $\tau = rh$, где $r = \text{const}$. Разностную задачу можно также записать в операторном виде:

$$L_h u_h = f_h, \quad (x, t) \in \bar{g}_h. \tag{8.12}$$

Значения сеточной функции u_i^j в узлах сетки $(x_i, t_j) \in \bar{g}_h$ приближенно заменяют значения искомой функции $U_i^j = U(x_i, t_j)$ в тех же узлах с погрешностями

$$\delta u_i^j = u_i^j - U_i^j. \tag{8.13}$$

Введем некоторое характерное значение этих погрешностей, например их максимальное по модулю значение на сетке

$$\delta u = \max_{i,j} |\delta u_i^j|.$$

Разностная схема (8.12) называется *сходящейся*, если при сгущении узлов сетки это значение погрешности стремится к нулю, т. е. если

$$\lim_{h \to 0} \delta u = 0. \tag{8.14}$$

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* $O(h^k)$.

Порядок точности схемы при наличии нескольких независимых переменных можно также оценивать по значениям шагов. 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.

Запишем уравнение (8.12) для погрешности решения на сетке: $\delta u_h = u_h - U_h$. Отсюда найдем $u_h = U_h + \delta u_h$. Подставляя это значение u_h в разностное уравнение (8.12), получаем

$$L_h \delta u_h = R_h, \quad R_h = f_h - L_h U_h. \tag{8.15}$$

Величина R_h называется *невязкой* (*погрешностью аппроксимации*) разностной схемы. Введем некоторую характерную величину невязки R, например

$$R = \max_{(x,t)\in \bar{g}_h} |R_h|. \tag{8.16}$$

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 (8.12) *аппроксимирует* исходную дифференциальную задачу (8.11), если при измельчении сетки невязка стремится к нулю, т. е. если

$$\lim_{\substack{h\to 0 \\ \tau\to 0}} R = 0. \tag{8.17}$$

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 $\tau < h^2$.

Разностная схема (8.12) 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, наклады-

ваются 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.

Теорема. *Если решение исходной дифференциальной задачи* (8.11) *существует, а разностная схема* (8.12) *устойчива и аппроксимирует задачу* (8.11) *на данном решении, то разностное решение сходится к точному.*

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 (8.7) и неявной схемы (8.8). Положим для простоты $a = 1$. Будем считать, что решение $U(x, t)$ дифференциальной задачи (8.6) 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) $(i = 1, 2, \ldots, I-1, j = 1, 2, \ldots, J-1)$ можно написать следующие соотношения:

$$\frac{u_i^{j+1} - u_i^j}{\tau} = \frac{\partial U(x_i, t_j)}{\partial t} + O(\tau), \tag{8.18}$$

$$\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2} = \frac{\partial^2 U(x_i, t_j)}{\partial x^2} + O(h^2). \tag{8.19}$$

Найдем погрешность аппроксимации R_i^j исходного уравнения (8.6) с помощью разностной схемы (8.7) для произвольного узла сетки (x_i, t_j):

$$R_i^j = \frac{u_i^{j+1} - u_i^j}{\tau} - \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{h^2}. \tag{8.20}$$

Подставим в это равенство соотношения (8.18), (8.19). При этом заметим, что значения $U(x_i, t_j)$ являются точ-

ным решением уравнения (8.6); поэтому

$$\frac{\partial U(x_i, t_j)}{\partial t} - \frac{\partial^2 U(x_i, t_j)}{\partial x^2} = 0. \tag{8.21}$$

Следовательно, максимальное значение невязки с учетом (8.18), (8.19), (8.21) имеет порядок

$$R = \max_{i,j} |R_i^j| = O(h^2 + \tau). \tag{8.22}$$

Аналогичную оценку невязки можно получить и для разностной схемы (8.8).

Таким образом, разностные схемы (8.7) и (8.8) аппроксимируют исходное дифференциальное уравнение (8.6) со вторым порядком по h и с первым порядком по τ. Начальное и граничные условия задачи (8.6) 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 (8.7). Рассмотрим решение v вспомогательной разностной задачи для уравнения теплопроводности с некоторой ненулевой функцией f в правой части и однородными начальными и граничными условиями. Запишем эту задачу в виде

$$\frac{v_i^{j+1} - v_i^j}{\tau} = \frac{v_{i+1}^j - 2v_i^j + v_{i-1}^j}{h^2} + f_i^j, \tag{8.23}$$

$$v_i^0 = 0, \quad v_0^j = 0, \quad v_I^j = 0. \tag{8.24}$$

Найдем из (8.23) значение вспомогательной сеточной функции v_i^{j+1} на верхнем слое:

$$v_i^{j+1} = \lambda v_{i-1}^j + (1 - 2\lambda) v_i^j + \lambda v_{i+1}^j + \tau f_i^j, \tag{8.25}$$

$$\lambda = \tau/h^2.$$

Допустим, что имеет место ограничение в виде неравенства

$$\lambda \leqslant 1/2. \tag{8.26}$$

Тогда $\lambda + |1 - 2\lambda| + \lambda = \lambda + 1 - 2\lambda + \lambda = 1$. Эти соотноше-

ния используем для оценки сеточного решения (8.25):

$$\max_{0\leqslant i\leqslant I}|v_i^{j+1}| = \max_{0\leqslant i\leqslant I}|\lambda v_{i-1}^j + (1-2\lambda)v_i^j + \lambda v_{i+1}^j + \tau f_i^j| \leqslant$$

$$\leqslant \max_{0\leqslant i\leqslant I}|v_i^j| + \tau \max_{1\leqslant i\leqslant I-1}|f_i^j|. \quad (8.27)$$

При $j=0$, учитывая (8.24), получаем

$$\max_{0\leqslant i\leqslant I}|v_i^1| \leqslant \tau f^*, \quad (8.28)$$

где f^* — некоторое характерное значение функции f в узлах сетки, например $f^* = \max\limits_{i,j}|f_i^j|$. Аналогично при $j=1$ из (8.27) с учетом (8.28) находим

$$\max_{0\leqslant i\leqslant I}|v_i^2| \leqslant \max_{0\leqslant i\leqslant I}|v_i^1| + \tau f^* \leqslant 2\tau f^*.$$

Следовательно, для некоторого характерного решения v^* в рассматриваемой области получаем оценку вида

$$v^* \leqslant J\tau f^* = cf^*. \quad (8.29)$$

В качестве v^* можно взять $v^* = \max\limits_{i,j}|v_i^j|$. Это неравенство означает устойчивость разностной схемы (8.7) по правой части при выполнении условия (8.26). Можно показать, что при нарушении этого условия схема (8.7) будет неустойчивой. Исследования устойчивости разностной схемы лишь по правой части в данном случае достаточно, поскольку начальное и граничные условия здесь аппроксимируются точно. Итак, явная схема (8.7) условно устойчива. Из аппроксимации и устойчивости следует ее сходимость со скоростью $O(h^2+\tau)$.

Для исследования устойчивости неявной разностной схемы (8.8) снова рассматривается по аналогии с (8.23) вспомогательная задача с однородными начальными и граничными условиями, но на этот раз неявная:

$$\frac{v_i^{j+1}-v_i^j}{\tau} = \frac{v_{i+1}^{j+1}-2v_i^{j+1}+v_{i-1}^{j+1}}{h^2} + f_i^j.$$

Отсюда для значений вспомогательной сеточной функции получается система линейных алгебраических уравнений

$$\lambda v_{i-1}^{j+1} - (1+2\lambda)v_i^{j+1} + \lambda v_{i+1}^{j+1} = -v_i^j - \tau f_i^j. \quad (8.30)$$

Эта система может быть решена методом прогонки. Безусловная устойчивость неявной схемы (8.8) обеспечивается выполнением условий устойчивости метода прогонки для системы (8.30).

Оценки устойчивости и сходимости разностных схем можно провести путем измельчения 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 ЭВМ 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

$$\frac{\partial U}{\partial t} + a\frac{\partial U}{\partial x} = F(x, t). \qquad (8.31)$$

Здесь a — скорость переноса, которую будем считать постоянной и положительной. Это соответствует переносу

(распространению возмущений) слева направо в положительном направлении оси x.

Характеристики уравнения (8.31) определяются соотношениями $x - at = C$. При постоянной a они являются прямыми линиями, которые в даном случае ($a > 0$) наклонены вправо (рис. 48).

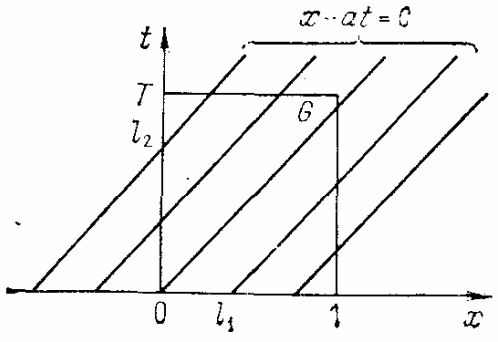


Рис. 48. Область решения

Расчетная область при решении уравнения (8.31) может быть как бесконечной, так и ограниченной. В первом случае, задавая начальное условие при $t = 0$:

$$U(x, 0) = \Phi(x), \quad (8.32)$$

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 $0 \leqslant t \leqslant T$, $0 \leqslant x \leqslant 1$; см. рис. 48). Начальное условие (8.32) 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

$$U(0, t) = \Psi(t). \quad (8.33)$$

Таким образом, задача состоит в решении уравнения (8.31) с начальным и граничным условиями (8.32) и (8.33) в ограниченной области G: $0 \leqslant t \leqslant T$, $0 \leqslant x \leqslant 1$.

Правильно (корректно) поставить данную задачу можно было бы с учетом вида решения уравнения (8.31), которое при $F(x, t) = 0$ имеет вид

$$U = H(x - at). \quad (8.34)$$

В этом легко убедиться, подставляя (8.34) в уравнение (8.31). Здесь H — произвольная дифференцируемая функция. Решение (8.34) 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. 48).

Можно также построить аналитическое решение задачи Коши для неоднородного уравнения (8.31). Заметим

лишь, что решение этой задачи меняется вдоль характеристики, а не является постоянным.

Рассмотрим разностные схемы для решения задачи (8.31)—(8.33). Построим в области G равномерную прямоугольную сетку с помощью прямых $x_i = ih$ $(i = 0, 1, \ldots, I)$, $t_j = j\tau$ $(j = 0, 1, \ldots, J)$. 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, ψ^j. Для построения разностной схемы необходимо выбрать шаблон. Примем его в виде правого нижнего уголка (рис. 49). При этом входящие в уравнение (8.31) производные аппроксимируются конечно-разностными соотношениями с использованием односторонних разностей:

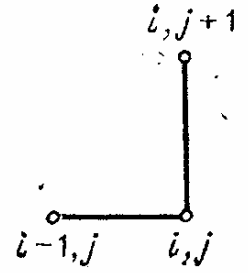


Рис. 49. Правый нижний уголок

$$\frac{u_i^{j+1} - u_i^j}{\tau} + a\frac{u_i^j - u_{i-1}^j}{h} = f_i^j. \qquad (8.35)$$

Решая разностное уравнение относительно единственного неизвестного значения u_i^{j+1} на $j+1$-м слое, получаем следующую разностную схему:

$$u_i^{j+1} = \lambda u_{i-1}^j + (1 - \lambda) u_i^j + \tau f_i^j,$$
$$\lambda = a\tau/h, \quad i = 1, 2, \ldots, I, \quad j = 0, 1, \ldots, J-1. \qquad (8.36)$$

Полученная схема — явная, поскольку значения сеточной функции в каждом узле верхнего слоя $t = t_{j+1}$ выражаются явно с помощью соотношений (8.36) через ранее найденные ее значения на предыдущем слое.

Для начала счета по схеме (8.36), т. е. для вычисления сеточной функции на первом слое, необходимы ее значения на слое $j = 0$. Они определяются начальным условием (8.32), которое записываем для сеточной функции:

$$u_i^0 = \varphi_i, \quad i = 0, 1, \ldots, I. \qquad (8.37)$$

Граничное условие (8.33) также записывается в сеточном виде:

$$u_0^j = \psi^j, \quad j = 1, 2, \ldots, J. \qquad (8.38)$$

Таким образом, решение исходной дифференциальной задачи (8.31)—(8.33) сводится к решению разностной задачи (8.36)—(8.38). Найденные значения сеточной

функции u_i^j принимаются в качестве значений искомой функции U в узлах сетки.

Алгоритм решения исходной задачи (8.31)—(8.33) с применением рассмотренной разностной схемы достаточно прост. На рис. 50 представлена его блок-схема. В соответствии с этим алгоритмом в памяти ЭВМ 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 хранить лишь значения сеточной функции на двух соседних слоях u_i^j, u_i^{j+1}. Рекомендуем читателю соответственным образом модифицировать представленный на рис. 50 алгоритм и построить новую блок-схему для двухслойной схемы.

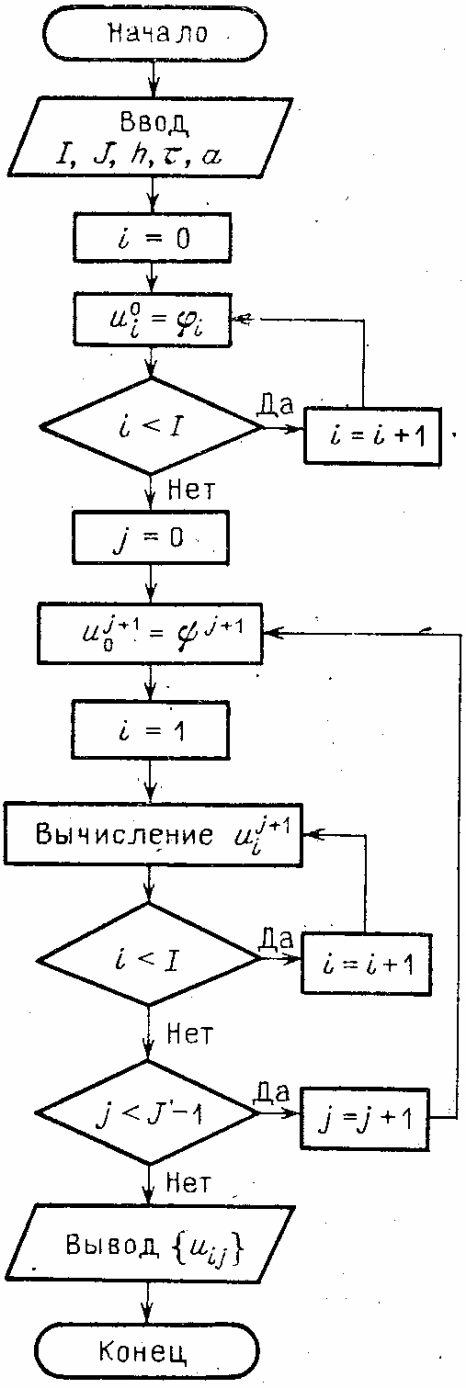


Рис. 50. Блок-схема решения линейного уравнения переноса

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

$$0 < \tau \leqslant h/a. \qquad (8.39)$$

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 (8.36) 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 (8.39) не выполняется, и схема (8.36) не сходится.

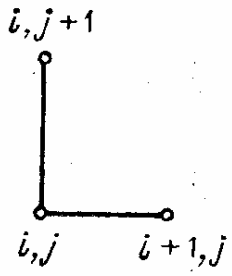


Рис. 51. Левый нижний уголок

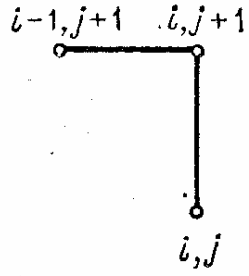


Рис. 52. Правый верхний уголок

Можно построить сходящуюся схему и для случая $a < 0$. В качестве шаблона для построения разностной схемы для уравнения (8.31) примем левый нижний уголок (рис. 51). Разностное уравнение в этом случае примет вид

$$\frac{u_i^{j+1} - u_i^j}{\tau} + a\,\frac{u_{i+1}^j - u_i^j}{h} = f_i^j. \qquad (8.40)$$

Эта схема является условно устойчивой (следовательно, сходящейся) при $a < 0$, если выполнено соотношение

$$\tau \leqslant -h/a, \qquad (8.41)$$

которое аналогично условию (8.39). При $a > 0$ эта схема не сходится.

При построении явной разностной схемы (8.36) производная $\partial U/\partial x$ в уравнении (8.31) аппроксимировалась с помощью значений сеточной функции на j-м слое; в результате получалось разностное уравнение (8.35), 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. 52), то получится неявная схема. Разностное уравнение примет вид

$$\frac{u_i^{j+1} - u_i^j}{\tau} + a\,\frac{u_i^{j+1} - u_{i-1}^{j+1}}{h} = f_i^j. \qquad (8.42)$$

Разрешая это уравнение относительно u_i^{j+1}, приходим к следующей разностной схеме:

$$u_i^{j+1} = \frac{u_i^j + \lambda u_{i-1}^{j+1} + \tau f_i^j}{1+\lambda}, \quad \lambda = \frac{a\tau}{h}. \tag{8.43}$$

Это двухслойная трехточечная схема первого порядка точности. Она безусловно устойчива. Хотя формально данная разностная схема строилась как неявная, практическая организация счета по ней проводится так же, как и для явных схем.

Действительно, в правую часть уравнения (8.43) 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 (8.38). По объему вычислений и логике программы (см. рис. 50) схема (8.43) аналогична схеме (8.36), однако безусловная устойчивость делает ее более удобной, поскольку исключается ограничение на величину шага.

Схему (8.36) можно применять для решения задачи Коши в неограниченной области, поскольку граничное условие (8.38) в этой схеме можно не использовать.

Рассмотрим еще одну разностную схему, которую построим на симметричном прямоугольном шаблоне (рис. 53). 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

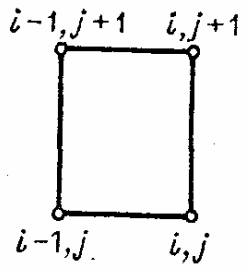


Рис. 53. Прямоугольник

$$\frac{1}{2}\left(\frac{u_{i-1}^{j+1} - u_{i-1}^j}{\tau} + \frac{u_i^{j+1} - u_i^j}{\tau}\right) +$$

$$+ \frac{1}{2}\left(\frac{u_i^j - u_{i-1}^j}{h} + \frac{u_i^{j+1} - u_{i-1}^{j+1}}{h}\right) = \bar{f}_i^j, \tag{8.44}$$

$$\bar{f}_i^j = f(x_i + h/2,\ t_j + \tau/2).$$

Данная двухслойная четырехточечная схема также формально построена как неявная. Однако из (8.44) можно выразить неизвестное значение u_i^{j+1} через остальные, которые предполагаются известными:

$$u_i^{j+1} = \frac{u_{i-1}^j (1+\lambda) + \left(u_i^j - u_{i-1}^{j+1}\right)(1-\lambda) + 2\tau \overline{f}_i^j}{1+\lambda}, \quad \lambda = \frac{a\tau}{h}. \tag{8.45}$$

Построенная схема имеет второй порядок точности. Она устойчива на достаточно гладких решениях.

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

$$\frac{\partial U}{\partial t} + a_1 \frac{\partial U}{\partial x} + a_2 \frac{\partial U}{\partial y} = F(x, y, t), \tag{8.46}$$

$$0 \leqslant x \leqslant 1, \quad 0 \leqslant y \leqslant 1, \quad 0 \leqslant t \leqslant T,$$

$$U(x, y, 0) = \Phi(x, y), \tag{8.47}$$

$$U(0, y, t) = \Psi_1(y, t), \quad U(x, 0, t) = \Psi_2(x, t). \tag{8.48}$$

Здесь $a_1 > 0$, $a_2 > 0$ — скорости переноса вдоль осей x, y; (8.47) — начальное условие при $t = 0$; (8.48) — граничные условия при $x = 0$, $y = 0$.

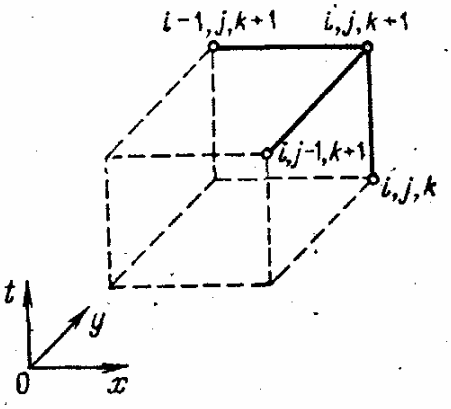


Рис. 54. Шаблон для двумерного уравнения

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: $x_i = ih_1$ $(i = 0, 1, \ldots, I)$, $y_j = jh_2$ $(j = 0, 1, \ldots, J)$, $t_k = k\tau$ $(k = 0, 1, \ldots, K)$. 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 (8.43). Шаблон изображен

на рис. 54, 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 (8.42) запишем разностное уравнение, аппроксимирующее дифференциальное уравнение (8.46):

$$\frac{u_{ij}^{k+1}-u_{ij}^{k}}{\tau}+a_1\frac{u_{ij}^{k+1}-u_{i-1,j}^{k+1}}{h_1}+a_2\frac{u_{ij}^{k+1}-u_{i,j-1}^{k+1}}{h_2}=f_{ij}^{k}. \quad (8.49)$$

Разрешим это уравнение относительно значения сеточной функции в узле $(i, j, k+1)$:

$$u_{ij}^{k+1}=\frac{u_{ij}^{k}+\lambda_1 u_{i-1,j}^{k+1}+\lambda_2 u_{i,j-1}^{k+1}+\tau f_{ij}^{k}}{1+\lambda_1+\lambda_2}, \quad (8.50)$$

$$\lambda_1=a_1\tau/h_1, \quad \lambda_2=a_2\tau/h_2.$$

Вычислительный алгоритм этой схемы аналогичен алгоритму одномерной схемы (8.43). Здесь также счет производится по слоям $k=1, 2, \ldots, K$. При $k=0$ используется начальное условие (8.47), которое нужно переписать в разностном виде:

$$u_{ij}^{0}=\varphi_{ij}. \quad (8.51)$$

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}, $u_{12}^{k+1}, \ldots, u_{1J}^{k+1}, u_{21}^{k+1}, \ldots, u_{IJ}^{k+1}$. На рис. 55 показана нумерация узлов, соответствующая данной последовательности вычислений на каждом временно́м слое. Точками отмечены расчетные узлы сетки, крестиками — граничные узлы, в которых значения сеточной функции задаются граничными условиями (8.48). Эти условия необходимо записать в сеточном виде:

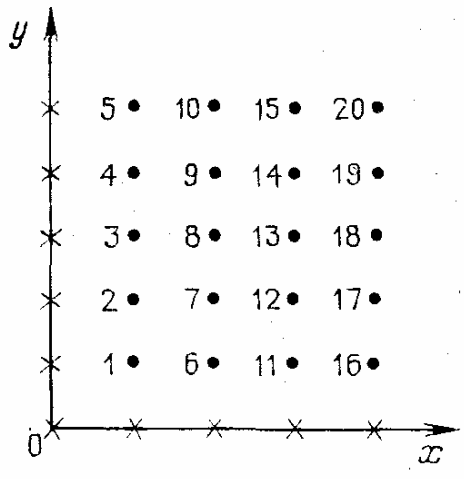


Рис. 55. Последовательность вычислений

$$u_{0j}^{k+1}=\psi_1(y_j, t_{k+1}), \quad u_{i0}^{k+1}=\psi_2(x_i, t_{k+1}). \quad (8.52)$$

При этом значение u_{00}^{k+1} в угловой точке $(x=0,\ y=0)$ в данной разностной схеме не используется.

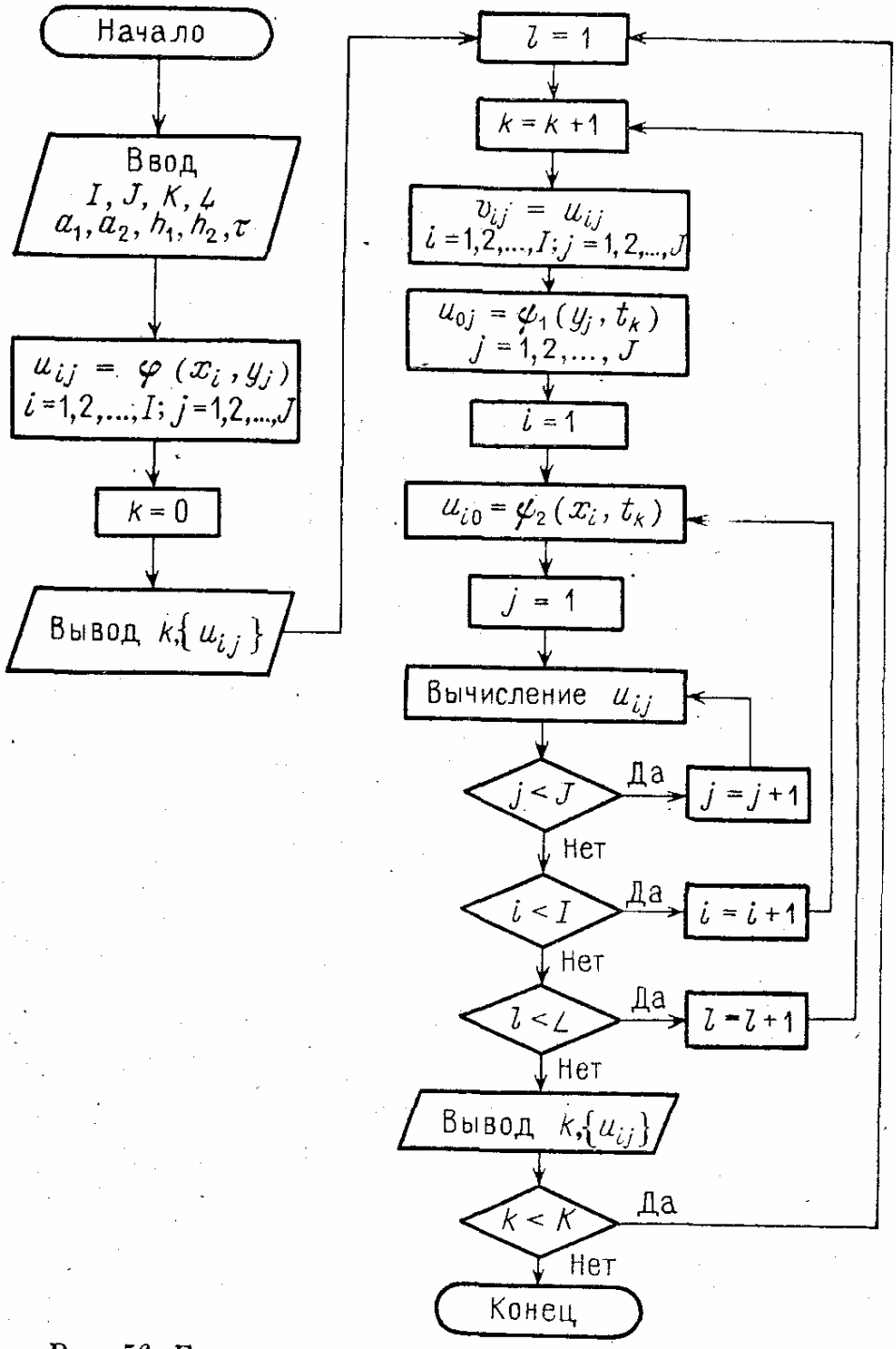


Рис. 56. Блок-схема решения двумерного уравнения

Блок-схема решения смешанной задачи (8.46)—(8.48) для двумерного уравнения переноса по схеме (8.50) с учетом сеточных начального и граничных условий (8.51) и (8.52) представлена на рис. 56. При этом неко-

торые блоки (вычисление начальных значений u_{ij}, значений на границе u_{0j}, пересылка $u_{ij} \to v_{ij}$) даны схематически, хотя каждый из них представляет циклический алгоритм.

В данной блок-схеме 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 (8.50), которая в принятых на блок-схеме 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. А 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

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} = 0. \tag{8.53}$$

Это однородное уравнение, т. е. его правая часть равна нулю, что указывает на отсутствие поглощения частиц (энергии) или источников.

Пусть в начальный момент времени ($t = 0$) решение уравнения (8.53) задано в виде

$$U(x, 0) = U_0(x). \tag{8.54}$$

В уравнении (8.53) 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 (8.53) в иной форме. Пусть $U = dx/dt$. Тогда уравнение примет вид

$$\frac{\partial U}{\partial t} + \frac{\partial U}{\partial x}\frac{dx}{dt} = 0.$$

Левая часть этого уравнения представляет полную производную по t сложной функции $U = U(x(t), t)$. Таким образом, мы приходим к системе уравнений

$$\frac{dx}{dt} = U, \quad \frac{dU}{dt} = 0, \tag{8.55}$$

которая равносильна уравнению (8.53). Решение этой системы (следовательно, и решение исходного уравнения) не меняется вдоль прямых

$$x = x_0 + U_0(x_0)t \tag{8.56}$$

и равно $U = U_0(x_0)$. Значение $U_0(x_0)$ соответствует начальному условию (8.54) в некоторой точке $x = x_0$.

Прямые линии (8.56) называются *характеристиками*. Вдоль характеристик уравнения вырождаются в некоторые соотношения между дифференциалами функции, называемые *соотношениями на характеристиках*.

Характеристики (8.56) квазилинейного уравнения (8.53), вообще говоря, не являются параллельными прямыми, как это было в случае линейного уравнения. Если переписать (8.56) 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 (8.53), (8.54) однозначно определено, поскольку через каждую точку полуплоскости $t > 0$ проходит одна характеристика, которая переносит в эту точку начальное значение. Такой случай показан на рис. 57.

Рассмотрим теперь другой случай. Пусть функция $U_0(x)$ монотонно убывает (или является такой хотя бы на небольшом участке). Тогда наклон характеристик при движении слева направо увеличивается (рис. 58), 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 неопределенны, поэтому уравнение теряет смысл. Следовательно, задачу нужно как-то доопределить, заменив в точках разрыва дифференциальные уравнения некоторыми конечными соотношениями.

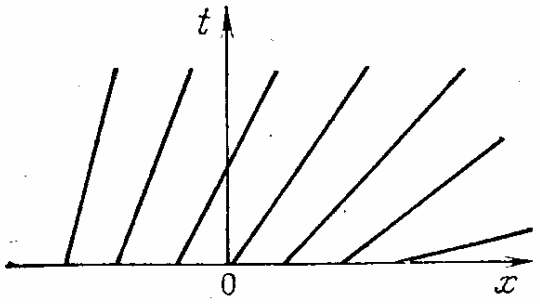


Рис. 57

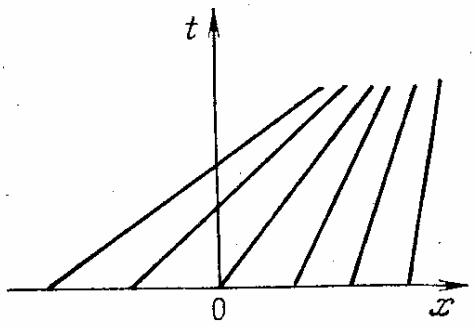


Рис. 58

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

$$\varphi'(t)=(U^-+U^+)/2, \quad U^->U^+. \tag{8.57}$$

Это соотношение на линии разрыва заменяет дифференциальное уравнение. Таким образом, решение задачи (8.53), (8.54), (8.57) 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 проводит-

ся 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 ЭВМ. 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. 59 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).

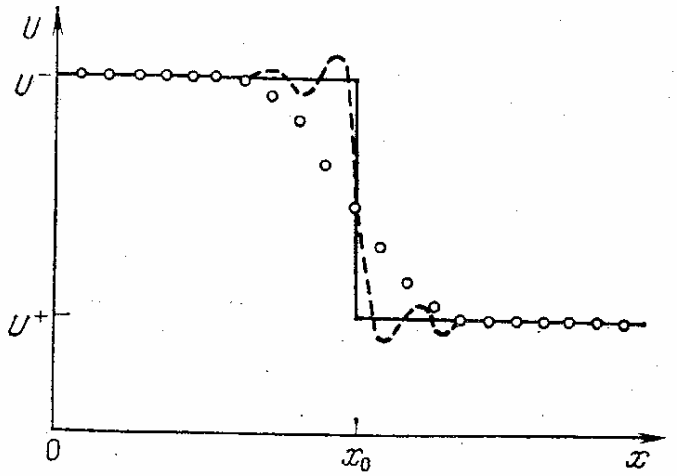


Рис. 59. Разрывное решение

Схемы, которые сохраняют монотонность решения разностной задачи, называются *монотонными*. В теории разностных схем доказывается следующий необходимый и достаточный признак монотонности линейной схемы.

Теорема. *Явная двухслойная разностная схема вида*

$$u_i^{j+1} = \alpha_1 u_{i-k}^j + \alpha_2 u_{i-k+1}^j + \ldots + \alpha_n u_{i+l}^j \qquad (8.58)$$

монотонна тогда и только тогда, когда $\alpha_1, \alpha_2, \ldots, \alpha_n$ — *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. 59 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 (8.36), (8.43) первого порядка точности обладают аппроксимационной вязкостью, а схема второго порядка (8.45) ею не обладает. Понятие аппроксимационной вязкости применимо только для линейных разностных схем вида (8.58).

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 (8.53) рассмотрим уравнение

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + \frac{\varepsilon^2}{2}\frac{\partial}{\partial x}\left(\frac{\partial U}{\partial x}\right)^2 = 0. \qquad (8.59)$$

Последний член этого уравнения есть искусственная вязкость, при этом параметр ε мал. Ясно, что при малом значении ε решения уравнений (8.53) и (8.59) при одинаковых начальных условиях будут близкими, если эти решения достаточно гладкие (вторая производная ограничена).

Рассмотрим теперь разрывное решение исходной задачи (8.53), (8.54). Пусть это решение представляет собой ступенчатую функцию (см. рис. 59)

$$U = \begin{cases} U^-, & x < at, \\ U^+, & x > at; \end{cases} \qquad (8.60)$$

$$a = (U^- + U^+)/2, \quad U^- > U^+. \qquad (8.61)$$

Это решение можно трактовать как ударную волну, движущуюся со скоростью a. При этом U^-, U^+ — некоторые постоянные. Легко убедиться в том, что функция (8.60) удовлетворяет как квазилинейному уравнению (8.53), так и соотношению (8.57), заменяющему на линии разрыва дифференциальное уравнение.

Построим решение уравнения (8.59). Будем искать это решение в виде

$$U_\varepsilon(x,\ t)=f(x-at). \tag{8.62}$$

На это решение можно наложить асимптотическое условие, которое состоит в том, что вдали от разрыва решение $U_\varepsilon(x,\ t)$ уравнения (8.59) и решение $U(x,\ t)$ уравнения (8.53), являющиеся гладкими функциями, близки, т. е.

$$f(x-at)\to U^{\pm},\quad x\to\pm\infty. \tag{8.63}$$

Решение (8.62) подставим в уравнение (8.59). При этом учтем, что функция $f(x-at)$ является сложной функцией одного аргумента $z=x-at$. Ее производные равны

$$\frac{\partial U}{\partial t}=\frac{df}{dz}\frac{\partial z}{\partial t}=-af',\quad \frac{\partial U}{\partial x}=\frac{df}{dz}\frac{\partial z}{\partial x}=f',$$

$$\frac{\partial}{\partial x}\left(\frac{\partial U}{\partial x}\right)^2=\frac{\partial}{\partial x}(f'^2)=2f'\frac{df'}{dz}\frac{\partial z}{\partial x}=2f'f''.$$

Подставляя эти выражения в уравнение (8.59), получаем следующее обыкновенное дифференциальное уравнение относительно искомой функции $f(x-at)$:

$$-af'+ff'+\varepsilon^2f'f''=0,$$

или

$$f'(\varepsilon^2f''+f-a)=0.$$

Приравнивая нулю каждый из сомножителей, получаем два значения функции f:

$$f_1=C_1,\quad f_2=a+C_2\sin[(x-at)/\varepsilon], \tag{8.64}$$

где C_1, C_2 — постоянные.

Из значений (8.64) с учетом (8.60), (8.61) можно построить решение, напоминающее «размазанную» ударную

волну (рис. 60), которое имеет вид

$$U_\varepsilon = \begin{cases} U^-, & \dfrac{x-at}{\varepsilon} \leqslant -\dfrac{\pi}{2}, \\ \dfrac{U^- + U^+}{2} = \dfrac{U^- - U^+}{2} \sin \dfrac{x-at}{\varepsilon}, & -\dfrac{\pi}{2} < \dfrac{x-at}{\varepsilon} < \dfrac{\pi}{2}, \\ U^+, & \dfrac{x-at}{t} \geqslant \dfrac{\pi}{2}. \end{cases} \tag{8.65}$$

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.

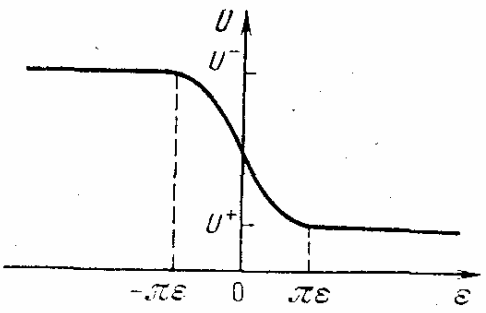


Рис. 60. Решение с искусственной вязкостью

Таким образом, вместо нахождения разрывного решения задачи (8.53), (8.54), (8.57) можно искать непрерывное решение уравнения (8.59) при малых значениях ε. А это уравнение решается с помощью однородных разностных схем. Следует при этом 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 (8.59) 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

$$u_i^{j+1} = u_i^j - \frac{\tau}{h} u_i^j (u_i^j - u_{i-1}^j) - \frac{\varepsilon^2 \tau}{2h^3}(u_{i+1}^j - u_{i-1}^j)(u_{i+1}^j - u_i^j + u_{i-1}^j). \tag{8.66}$$

Эта явная схема условно устойчива при выполнении неравенства

$$\tau \leqslant h/U(x, t), \tag{8.67}$$

в котором роль скорости распространения возмущения a (для линейного уравнения) играет сама функция U. Разностная схема (8.66) 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 такая 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, т. е. когда в качестве искомых параметров принимаются масса, энергия, количество движения и т. п., а эти уравнения выражают законы сохранения этих параметров. Такая форма записи уравнений называется *дивергентной*.

Формально квазилинейное уравнение переноса можно также записать в дивергентной форме:

$$\frac{\partial U}{\partial t} + \frac{\partial}{\partial x}\left(\frac{U^2}{2}\right) = 0. \tag{8.68}$$

Проинтегрируем это уравнение по ячейке $x_{i-1} \leqslant x \leqslant x_i$, $t_j \leqslant t \leqslant t_{j+1}$:

$$\int_{x_{i-1}}^{x_i} dx \int_{t_j}^{t_{j+1}} \frac{\partial U}{\partial t}\, dt + \int_{x_{i-1}}^{x_i} dx \int_{t_j}^{t_{j+1}} \frac{\partial}{\partial x}\left(\frac{U^2}{2}\right) dt = 0,$$

или

$$\int_{x_{i-1}}^{x_i} \left[U^{j+1}(x) - U^j(x)\right] dx + \frac{1}{2}\int_{t_j}^{t_{j+1}} \left[U_i^2(t) - U_{i-1}^2(t)\right] dt = 0. \tag{8.69}$$

Последнее уравнение представляет собой точное интегральное уравнение для данной ячейки. Обычно при исследовании физических процессов оно выражает закон сохранения.

Аналогичное интегральное уравнение можно получить для всей расчетной области $x_0 \leqslant x \leqslant x_I$, $0 \leqslant t \leqslant t_J$, если проинтегрировать уравнение (8.68) по этой области. Оно имеет вид

$$\int_{x_0}^{x_I} \left(U^J - U^0\right) dx + \frac{1}{2}\int_0^{t_J} \left(U_I^2 - U_0^2\right) dt = 0. \tag{8.70}$$

Уравнения (8.69) и (8.70) можно трактовать как физические законы сохранения. При этом, если просуммировать уравнение (8.69) по всем ячейкам, получается уравнение (8.70) 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 (8.69). Воспользуемся для простоты формулой прямоугольников, причем узлы предполагаем совпадающими с узлами рассматриваемой разностной сетки. Окончательно получим разностную схему вида

$$\frac{u_i^{j+1} - u_i^j}{\tau} + \frac{\left(u_i^j\right)^2 - \left(u_{i-1}^j\right)^2}{2h} = 0.$$

Отсюда можно найти значение искомой функции на верхнем слое с помощью решения на нижнем слое. Следовательно, это явная схема. Она обладает свойством консер-

вативности. 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 требования 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 $u(x, t)$, $v(x, t)$:

$$\begin{aligned} a_{11}\frac{\partial u}{\partial t} + a_{12}\frac{\partial v}{\partial t} + b_{11}\frac{\partial u}{\partial x} + b_{12}\frac{\partial v}{\partial x} &= f_1(x, t, u, v), \\ a_{21}\frac{\partial u}{\partial t} + a_{22}\frac{\partial v}{\partial t} + b_{21}\frac{\partial u}{\partial x} + b_{22}\frac{\partial v}{\partial x} &= f_2(x, t, u, v). \end{aligned} \tag{8.71}$$

Коэффициенты a_{mn}, b_{mn} $(m, n = 1, 2)$ этой системы переменные и зависят от x, t, u, v. Введем следующие обозначения: $U = \{u, v\}$ — искомый вектор; $F = \{f_1, f_2\}$ — вектор правой части; A, B — матрицы коэффициентов:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

Запишем систему уравнений (8.71) в векторном виде:

$$A\frac{\partial U}{\partial t} + B\frac{\partial U}{\partial x} = F. \tag{8.72}$$

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 (8.71), 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

$$\det C = \begin{vmatrix} a_{11}\alpha - b_{11}\beta & a_{12}\alpha - b_{12}\beta \\ a_{21}\alpha - b_{21}\beta & a_{22}\alpha - b_{22}\beta \end{vmatrix} \tag{8.73}$$

является *квадратичной формой* относительно α, β, т. е.

$$\det C = Q(\alpha, \beta) = q_1\alpha^2 + q_2\alpha\beta + q_3\beta^2, \tag{8.74}$$

где коэффициенты q_1, q_2, q_3 легко выразить через элементы матриц A, B, раскрывая определитель (8.73).

Система уравнений с частными производными первого порядка (8.71) называется *гиперболической*, если квадратичная форма (8.74) разлагается на вещественные линейные множители:

$$Q(\alpha, \beta) = (\mu_1\alpha - \nu_1\beta)(\mu_2\alpha - \nu_2\beta),$$

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. 61).

Заметим, что в случае системы уравнений (8.71) 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 (8.71) зависят от x, t, u, v, характеристики могут существовать в одной части плоскости (x, t) и отсутствовать в другой. Следовательно, гиперболичность системы (8.71) 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).

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 $A_0, A_1, \ldots, A_n$ (рис. 62). В данном случае принято $n = 4$.

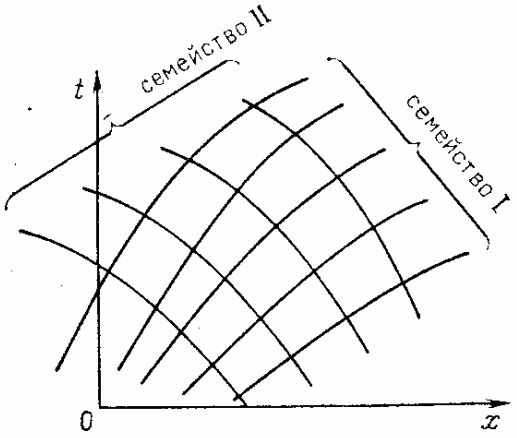


Рис. 61. Характеристики

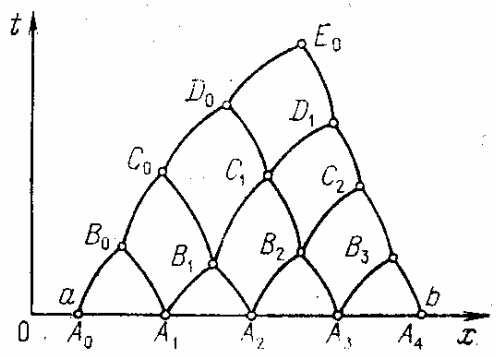


Рис. 62

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, заменяющих 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, v. 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.

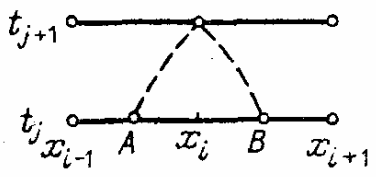


Рис. 63

Геометрическая интерпретация сеточно-характеристического метода показана на рис. 63. 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

$$\frac{\partial^2 U}{\partial t^2} = a^2 \frac{\partial^2 U}{\partial x^2}. \tag{8.75}$$

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 (8.75) 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 (8.75) является задача Коши: в начальный момент времени задаются два условия (количество условий равно порядку входящей в уравнение производной по t):

$$U|_{t=0} = U(x, 0) = \varphi(x), \quad \partial U/\partial t|_{t=0} = \psi(x). \tag{8.76}$$

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

$$U|_{x=0} = 0, \quad U|_{x=l} = 0. \tag{8.77}$$

Рассмотрим некоторые разностные схемы для решения задачи (8.75)—(8.77). Простейшей является явная трехслойная схема крест (шаблон показан на рис. 64). Заменим в уравнении (8.75) вторые производные искомой

функции U по t и x 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:

$$u_i^{j+1} = 2(1-\lambda)u_i^j + \lambda(u_{i+1}^j + u_{i-1}^j) - u_i^{j-1},$$
$$\lambda = a^2\tau^2/h^2. \qquad (8.78)$$

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 (8.78) можно лишь для второго слоя, а решения на нулевом и первом слоях должны быть известны. Они находятся с помощью начальных условий (8.76).

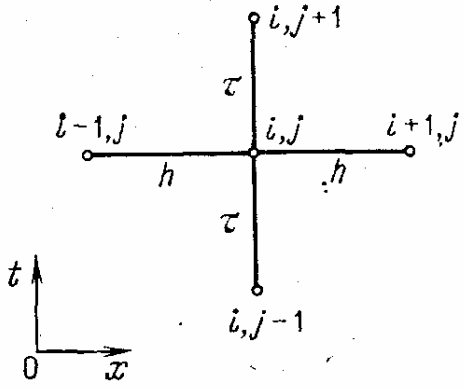


Рис. 64. Шаблон явной схемы

На нулевом слое имеем

$$u_i^0 = \varphi(x_i), \quad i = 0, 1, \ldots, I, \qquad (8.79)$$

Для получения решения на первом слое воспользуемся вторым начальным условием (8.76). Производную $\partial U/\partial t$ заменим конечно-разностной аппроксимацией. В простейшем случае полагают

$$\left.\frac{\partial U}{\partial t}\right|_{t=0} \approx \frac{u_i^1 - u_i^0}{\tau} \approx \psi(x_i). \qquad (8.80)$$

Из этого соотношения можно найти значения сеточной функции на первом временном слое:

$$u_i^1 = u_i^0 + \tau\psi(x_i), \quad i = 0, 1, \ldots, I. \qquad (8.81)$$

Отметим, что аппроксимация начального условия в виде (8.80) ухудшает аппроксимацию исходной дифференциальной задачи: погрешность аппроксимации становится порядка $O(h^2 + \tau)$, т. е. первого порядка по τ, хотя сама

схема (8.78) имеет второй порядок аппроксимации по h и τ. Положение можно исправить, если вместо (8.81) взять более точное представление

$$u_i^1 = u_i^0 + \tau \left.\frac{\partial U}{\partial t}\right|_{t=0} + \frac{\tau^2}{2} \left.\frac{\partial^2 U}{\partial t^2}\right|_{t=0}. \qquad (8.82)$$

Вместо $\partial U/\partial t$ нужно взять $\psi(x)$. А выражение для второй производной можно найти с использованием уравнения (8.75) и первого начального условия (8.76). Получим

$$\left.\frac{\partial^2 U}{\partial t^2}\right|_{t=0} = a^2 \left.\frac{\partial^2 U}{\partial x^2}\right|_{t=0} = a^2 \frac{d^2\varphi}{dx^2}.$$

Тогда (8.82) принимает вид

$$u_i^1 = u_i^0 + \tau\psi(x_i) + \frac{a^2\tau^2}{2}\varphi''(x_i), \quad i = 0, 1, \ldots, I, \qquad (8.83)$$

Разностная схема (8.78) с учетом (8.83) обладает погрешностью аппроксимации порядка $O(h^2 + \tau^2)$.

При решении смешанной задачи с граничными условиями вида (8.77), 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

$$\partial U/\partial x|_{x=l} = 0. \qquad (8.84)$$

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 (8.84) аппроксимировать со вторым порядком.

Рассмотренная разностная схема (8.78) решения задачи (8.75)—(8.77) условно устойчива. Необходимое и достаточное условие устойчивости имеет вид

$$a\tau/h < 1. \qquad (8.85)$$

Следовательно, при выполнении этого условия и с учетом аппроксимации схема (8.78) 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.

Блок-схема решения задачи (8.75)—(8.77) с помощью данной явной разностной схемы приведена на рис. 65. Здесь приведен простейший алгоритм, когда все значения сеточной функции, образующие двумерный массив, по мере вычисления хранятся в памяти ЭВМ, 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. 56).

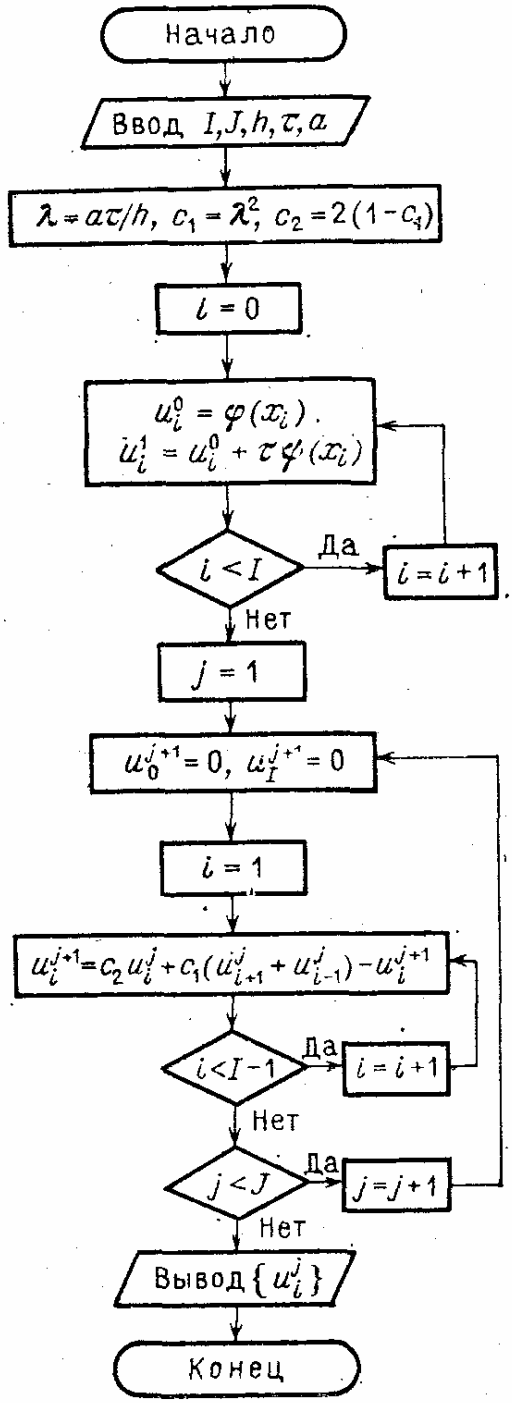


Рис. 65. Блок-схема решения волнового уравнения

Существуют и другие разностные схемы решения волнового уравнения. В частности, иногда удобнее использовать неявные схемы, чтобы избавиться от ограничений на величину шага, налагаемых условием (8.85). Эти схемы обычно абсолютно устойчивы, однако алгоритм решения задачи и программа на ЭВМ усложняются.

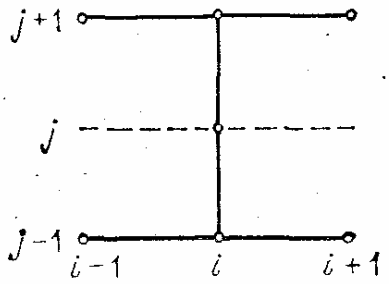


Рис. 66. Шаблон неявной схемы

Построим простейшую неявную схему. Вторую производную по t в уравнении (8.75) аппроксимируем, как и ранее, по трехточечному шаблону с помощью значений

сеточной функции на слоях $j-1$, j, $j+1$. Производную по x заменяем полусуммой ее аппроксимации на $j+1$-м и $j-1$-м слоях (рис. 66):

$$\frac{u_i^{j+1}-2u_i^j+u_i^{j-1}}{\tau^2}=\frac{a^2}{2}\left(\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{h^2}+\frac{u_{i+1}^{j-1}-2u_i^{j-1}+u_{i-1}^{j-1}}{h^2}\right). \tag{8.86}$$

Из этого разностного соотношения можно получить систему уравнений относительно неизвестных значений сеточной функции на $j+1$-м слое ($j=1, 2, \ldots$):

$$\begin{gathered}\lambda u_{i-1}^{j+1}-(1+2\lambda)\,u_i^{j+1}+\lambda u_{i+1}^{j+1}=(1+2\lambda)\,u_i^{j-1}-\\ -\lambda\left(u_{i+1}^{j-1}+u_{i-1}^{j-1}\right)-2u_i^j,\\ \lambda=a^2\tau^2/h^2,\quad i=1, 2, \ldots, I-1.\end{gathered} \tag{8.87}$$

Полученная неявная схема устойчива и сходится со скоростью $O(h^2+\tau^2)$. Систему линейных алгебраических уравнений (8.87) можно, в частности, решать методом прогонки. К этой системе следует добавить разностные начальные и граничные условия. В частности, выражения (8.79), (8.81) или (8.83) 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 соответственно имеют вид

$$\frac{\partial^2 U}{\partial t^2}=a^2\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}\right),$$

$$\frac{\partial^2 U}{\partial t^2}=a^2\left(\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}+\frac{\partial^2 U}{\partial z^2}\right).$$

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, п. 2) уже были построены разностные схемы решения смешан-

ной задачи для одномерного уравнения теплопроводности:

$$\frac{\partial U}{\partial t} = a\frac{\partial^2 U}{\partial x^2}, \quad 0 \leqslant x \leqslant 1, \quad t > 0, \tag{8.88}$$

$$U(x, 0) = \varphi(x), \tag{8.89}$$

$$U(0, t) = \psi_1(t), \quad U(1, t) = \psi_2(t). \tag{8.90}$$

Были получены две двухслойные схемы — явная и неявная. В явной схеме значения сеточной функции $u_i^{j+1} = u(x_i, t_{j+1})$ на верхнем $j+1$-м слое вычислялись с помощью решения на нижнем слое:

$$\begin{gathered} u_i^{j+1} = \lambda u_{i+1}^j + (1 - 2\lambda) u_i^j + \lambda u_{i-1}^j, \\ \lambda = a\tau/h^2, \quad i = 1, 2, \ldots, I-1. \end{gathered} \tag{8.91}$$

Данная схема сходится к решению исходной задачи со скоростью $O(h^2 + \tau)$. Она устойчива при выполнении условия

$$\lambda = a\tau/h^2 \leqslant 1/2. \tag{8.92}$$

При $\lambda = 1/2$ получается особенно простой вид рекуррентных соотношений (8.91):

$$u_i^{j+1} = (u_{i+1}^j + u_{i-1}^j)/2. \tag{8.93}$$

Условие устойчивости (8.92) накладывает ограничение на шаг по t при выбранном значении h, что характерно для явных схем.

Построим простейшую неявную схему. Производную $\partial^2 U/\partial x^2$ аппроксимируем на $j+1$-м слое:

$$\frac{\partial^2 U}{\partial x^2} \approx \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2}.$$

В этом случае получается трехдиагональная система линейных алгебраических уравнений относительно значений сеточной функции в узлах верхнего слоя, которая имеет вид

$$\lambda u_{i-1}^{j+1} - (1 + 2\lambda) u_i^{j+1} + \lambda u_{i+1}^{j+1} = -u_i^j, \quad i = 1, 2, \ldots, I-1. \tag{8.94}$$

Эта система может быть решена методом прогонки. При этом разностное решение сходится к точному со вторым порядком по h и с первым порядком по τ. Схема (8.94) безусловно устойчива.

Выражения (8.91) и (8.94) определяют значения сеточной функции во внутренних узлах, а решение на границе находится из граничных условий, которые зависят от конкретной постановки задачи. В частности, если граничные условия имеют вид (8.90), то на каждом слое

$$u_0^j = \psi_1(t_j), \quad u_J^j = \psi_2(t_j). \tag{8.95}$$

В граничные условия может также входить производная искомой функции (температуры). Например, если конец стержня $x = 0$ теплоизолирован, то условие имеет вид

$$\partial U/\partial x|_{x=0} = 0. \tag{8.96}$$

В этом случае, как и при решении волнового уравнения, данное условие нужно записывать в разностном виде.

Перейдем к построению разностных схем для уравнения теплопроводности с двумя пространственными переменными. Положим для простоты $a = 1$. Тогда это уравнение можно записать в виде

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}. \tag{8.97}$$

Пусть при $t = 0$ начальное условие задано в виде

$$U(x, y, 0) = \varphi(x, y). \tag{8.98}$$

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 (8.97), решаются в ограниченной области. Тогда, кроме начального условия (8.98), 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. 67), то нужно задавать граничные условия на его боковых гранях. Начальное условие (8.98) задано на нижнем основании параллелепипеда.

Введем простейшую сетку с ячейками в виде прямоугольных параллелепипедов, для чего проведем три семейства плоскостей: $x_i = ih_1$ $(i = 0, 1, \ldots, I)$, $y_j = jh_2$ $(j = 0, 1, \ldots, J)$, $t_k = k\tau$ $(k = 0, 1, \ldots, K)$. Значения сеточной функции в узлах (x_i, y_j, t_k) обозначим символом u_{ij}^k. Используя эти значения, можно построить разностные

схемы для уравнения (8.97). Рассмотренные выше схемы легко обобщаются на двумерный случай.

Построим явную разностную схему, шаблон которой изображен на рис. 68. 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:

$$u_{ij}^{k+1}=(1-2\lambda_1-2\lambda_2)u_{ij}^{k}+\lambda_1\left(u_{i+1,j}^{k}+u_{i-1,j}^{k}\right)+$$
$$+\lambda_2\left(u_{i,j+1}^{k}+u_{i,j-1}^{k}\right),\quad \lambda_1=\tau/h_1^2,\quad \lambda_2=\tau/h_2^2. \qquad (8.99)$$

Условие устойчивости имеет вид

$$\lambda_1+\lambda_2=\tau/h_1^2+\tau/h_2^2\leqslant 1/2 \qquad (8.100)$$

При $\lambda_1+\lambda_2=1/2$ получается особенно простой вид схемы (8.99):

$$u_{ij}^{k+1}=\lambda_1\left(u_{i+1,j}^{k}+u_{i-1,j}^{k}\right)+\lambda_2\left(u_{i,j+1}^{k}+u_{i,j-1}^{k}\right). \qquad (8.101)$$

Полученная схема сходится со скоростью $O\left(h_1^2, h_2^2, \tau\right)$.

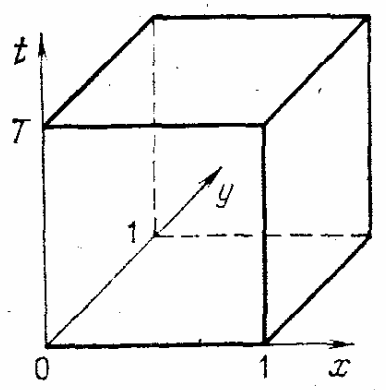


Рис. 67. Расчетная область

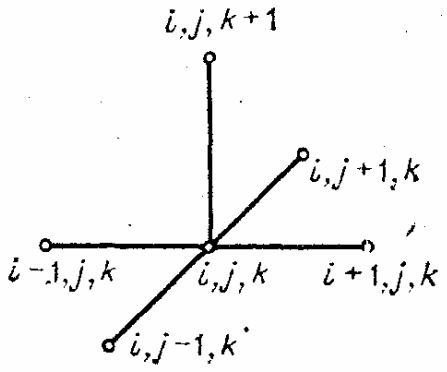


Рис. 68. Шаблон двумерной схемы

Формулы (8.99) или (8.101) представляют рекуррентные соотношения для последовательного вычисления сеточной функции во внутренних узлах слоев $k=1, 2, \ldots$ $\ldots, K$. На нулевом слое используется начальное условие (8.98), которое записывается в виде

$$u_{ij}^{0}=\varphi(x_i, y_j). \qquad (8.102)$$

Значения $u_{0j}^k, u_{1j}^k, u_{i0}^k, u_{iJ}^k$ в граничных узлах вычисляются с помощью граничных условий.

Блок-схема решения смешанной задачи для двумерного уравнения теплопроводности изображена на рис. 69. 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. 56).

Можно построить абсолютно устойчивую неявную схему для решения уравнения (8.97), аналогичную схеме (8.94) для одномерного уравнения теплопроводности. Аппроксимируя в (8.97) вторые производные по пространственным переменным на $k+1$-м слое, получаем следующее разностное уравнение:

$$\frac{u_{ij}^{k+1} - u_{ij}^k}{\tau} = \frac{u_{i+1,j}^{k+1} - 2u_{ij}^{k+1} + u_{i-1,j}^{k+1}}{h_1^2} + \frac{u_{i,j+1}^{k+1} - 2u_{ij}^{k+1} + u_{i,j-1}^{k+1}}{h_2^2}. \tag{8.103}$$

Это уравнение можно записать в виде системы линейных алгебраических уравнений относительно значений сеточной функции на каждом слое:

$$\lambda_1 (u_{i-1,j}^{k+1} + u_{i+1,j}^{k+1}) - (1 + 2\lambda_1 + 2\lambda_2) u_{ij}^{k+1} + \\ + \lambda_2 (u_{i,j-1}^{k+1} + u_{i,j+1}^{k+1}) = -u_{ij}^k, \quad \lambda_1 = \tau/h_1^2, \quad \lambda_2 = \tau/h_2^2, \tag{8.104}$$

$$i = 1, 2, \ldots, I-1, \; j = 1, 2, \ldots, J-1.$$

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 (8.98), представленного в виде (8.102).

Система (8.104), полученная для двумерного уравнения теплопроводности, имеет более сложный вид, чем аналогичная система (8.94) 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 (8.99) является жесткое ограничение на шаг по времени τ, вытекающее из условия

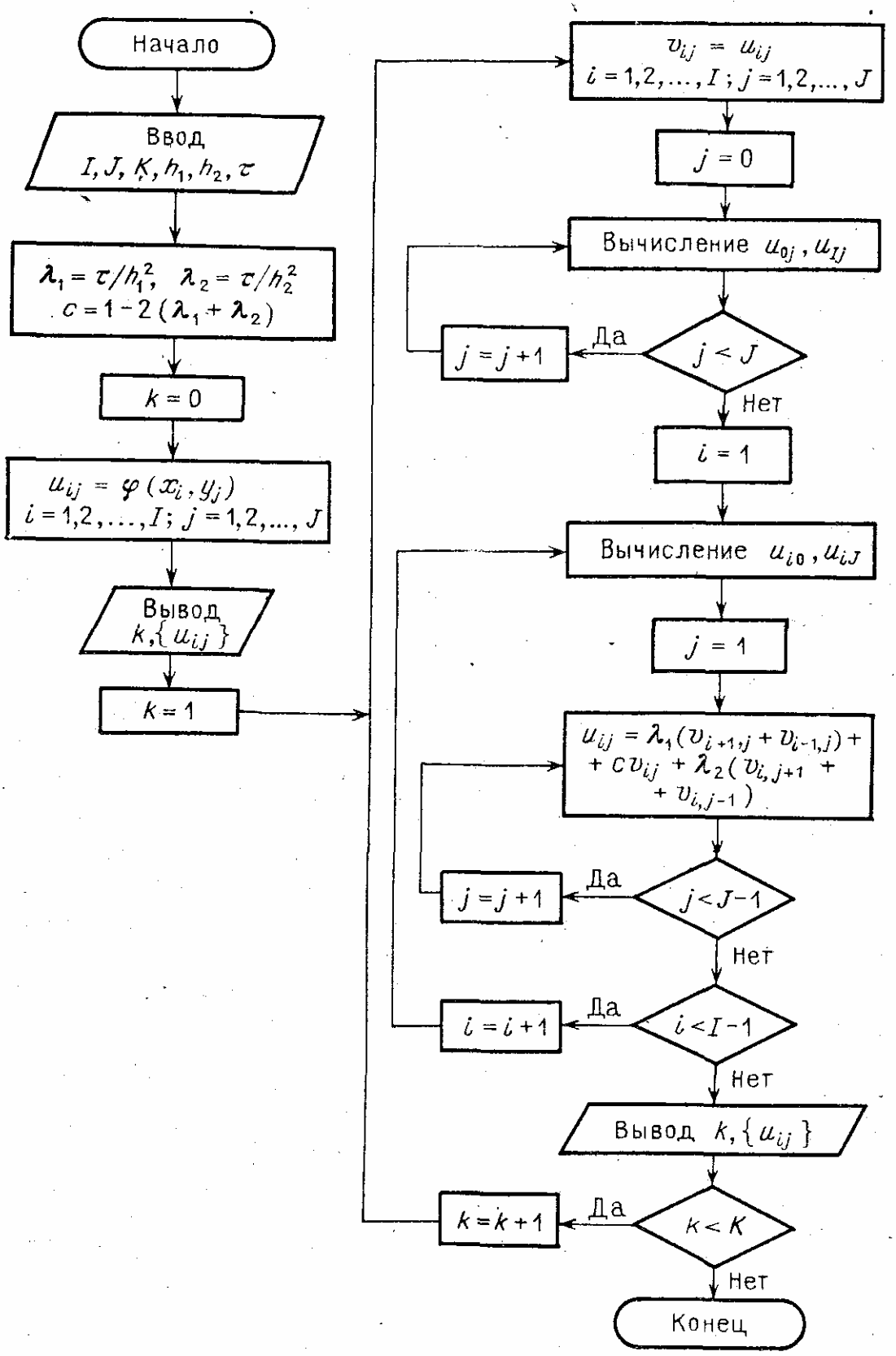


Рис. 69. Блок-схема решения двумерного уравнения теплопроводности.

(8.100). 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. На первом из них вторая производная по одной из координат, например $\partial^2 U/\partial x^2$, аппроксимируется на промежуточном слое $k+1/2$, т. е. используется неявная аппроксимация; в этом случае $\partial^2 U/\partial y^2$ аппроксимируется на слое k, т. е. явно. На втором полушаге наоборот, неявная аппроксимация используется только по направлению y.

Соответствующая разностная схема для двумерного уравнения теплопроводности имеет вид

$$\frac{u_{ij}^{k+1/2}-u_{ij}^{k}}{\tau/2}=\frac{u_{i+1,j}^{k+1/2}-2u_{ij}^{k+1/2}+u_{i-1,j}^{k+1/2}}{h_1^2}+\frac{u_{i,j+1}^{k}-2u_{ij}^{k}+u_{i,j-1}^{k}}{h_2^2}, \tag{8.105}$$

$$\frac{u_{ij}^{k+1}-u_{ij}^{k+1/2}}{\tau/2}=\frac{u_{i+1,j}^{k+1/2}-2u_{ij}^{k+1/2}+u_{i-1,j}^{k+1/2}}{h_1^2}+\frac{u_{i,j+1}^{k+1}-2u_{ij}^{k+1}+u_{i,j-1}^{k+1}}{h_2^2}. \tag{8.106}$$

Таким образом, вместо разностного уравнения (8.103) в чисто неявной схеме мы получили два уравнения, каждое из которых, по существу, соответствует неявной схеме по одному из координатных направлений.

Уравнения (8.105), (8.106) можно переписать в виде систем линейных алгебраических уравнений относительно значений искомых функций соответственно в узлах

$k+1/2$-го и $k+1$-го слоев:

$$\lambda_1 u_{i-1,j}^{k+1/2} - (1+2\lambda_1) u_{ij}^{k+1/2} + \lambda_1 u_{i+1,j}^{k+1/2} = \\ = (2\lambda_2 - 1) u_{ij}^{k} - \lambda_2 \left(u_{i,j+1}^{k} + u_{i,j-1}^{k}\right), \quad (8.107)$$

$$\lambda_2 u_{i,j-1}^{k+1} - (1+2\lambda_2) u_{ij}^{k+1} + \lambda_2 u_{i,j+1}^{k+1} = \\ = (2\lambda_1 - 1) u_{ij}^{k+1/2} - \lambda_1 \left(u_{i+1,j}^{k+1/2} + u_{i-1,j}^{k+1/2}\right), \quad (8.108)$$

$$\lambda_1 = \tau/(2h_1^2), \quad \lambda_2 = \tau/(2h_2^2),$$

$$i = 1, 2, \ldots, I, \quad j = 1, 2, \ldots, J.$$

К этим системам уравнений необходимо добавить начальные условия в виде (8.102), а также граничные условия на каждом из этих дробных по времени шагов.

Матрицы систем (8.107) и (8.108) трехдиагональные, и для решения этих систем может быть использован метод прогонки. При этом сначала необходимо решить систему уравнений (8.107), 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 (8.108).

Заметим, что диагональные элементы матриц систем (8.107) и (8.108) 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 $O\left(h_1^2 + h_2^2 + \tau^2\right)$.

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 разност-

ные схемы не аппроксимируют исходное уравнение. Здесь имеет место лишь суммарная аппроксимация на слоях с целыми номерами. Погрешности аппроксимации 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

$$\frac{\tilde{u}_{ij} - u_{ij}^k}{\tau} = \frac{\tilde{u}_{i+1,j} - 2\tilde{u}_{ij} + \tilde{u}_{i-1,j}}{h_1^2},$$

$$\frac{u_{ij}^{k+1} - \tilde{u}_{ij}}{\tau} = \frac{u_{i,j+1}^{k+1} - 2u_{ij}^{k+1} + u_{i,j-1}^{k+1}}{h_2^2},$$

$$i = 1, 2, \ldots, I, \; J = 1, 2, \ldots, J.$$

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 $O\left(h_1^2 + h_2^2 + \tau\right)$.

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 ре-

зультатов с учетом процессов переноса. Изложение вопросов, связанных с построением указанных схем, можно найти в специальной литературе.

4. Уравнение Лапласа. Многие стационарные физические задачи (исследования потенциальных течений жидкости, определение формы нагруженной мембраны, задачи теплопроводности и диффузии в стационарных случаях и др.) сводятся к решению уравнения Пуассона вида

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = F(x, y, z). \qquad (8.109)$$

Если $F(x, y, z) = 0$, то уравнение (8.109) называется *уравнением Лапласа*. Для простоты будем рассматривать двумерное уравнение Лапласа

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0, \qquad (8.110)$$

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

$$U(x, y)|_L = \varphi(x, y). \qquad (8.111)$$

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 (8.110), (8.111), является их сведение к решению некоторой фиктивной нестационарной задачи (гиперболической или параболической), найденное решение которой при достаточно больших значениях t близко к решению исходной задачи. Такой способ решения называется *методом установления*.

Поскольку решение $U(x, y)$ уравнения (8.110) не зависит от времени, то можно в это уравнение добавить равный нулю (при точном решении) член $\partial U/\partial t$. Тогда уравнение (8.110) примет вид

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}. \qquad (8.112)$$

Это — известное нам уравнение теплопроводности, для которого в п. 3 уже строились разностные схемы. Остается только задать начальное условие. Его можно принять практически в произвольном виде, согласованном с граничными условиями. Положим

$$U|_{t=0} = \psi(x, y). \qquad (8.113)$$

Граничное условие (8.111) при этом остается стационарным, т. е. не зависящим от времени.

Процесс численного решения уравнения (8.112) с условиями (8.113), (8.111) состоит в переходе при $t \to \infty$ от произвольного значения (8.113) 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 (8.112), (8.113), (8.111), 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.

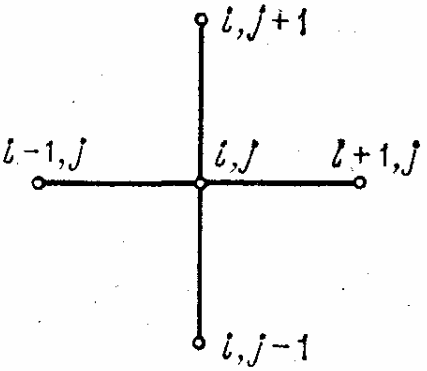


Рис. 70. Шаблон для уравнения Лапласа

Для решения задачи Дирихле можно также построить разностную схему путем аппроксимации уравнения (8.110). 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_i) заменим значениями сеточной функции u_{ij}. Тогда, аппроксимируя в уравнении (8.110) вторые производные с помощью отношений конечных разностей, получим разностное уравнение (шаблон изображен на рис. 70):

$$\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h^2} = 0. \qquad (8.114)$$

Данное уравнение можно представить в виде системы линейных алгебраических уравнений относительно значений сеточной функции в узлах. Эту систему можно записать в виде

$$u_{i+1,\,j} + u_{i-1,\,j} + u_{i,\,j+1} + u_{i,\,j-1} - 4u_{ij} = 0, \qquad (8.115)$$
$$i = 1,\ 2,\ \ldots,\ I-1, \quad j = 1,\ 2,\ \ldots,\ J-1.$$

Значения сеточной функции в узлах, расположенных на границе расчетной области, могут быть найдены из граничного условия (8.111):

$$u_{0j} = \varphi(x_0,\ y_j), \quad u_{Ij} = \varphi(x_I,\ y_j), \quad j = 0,\ 1,\ \ldots,\ J;$$
$$u_{i0} = \varphi(x_i,\ y_0), \quad u_{iJ} = \varphi(x_i,\ y_J), \quad i = 0,\ 1,\ \ldots,\ I.$$

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 (8.115). Каждое уравнение системы (за исключением тех, которые соответствуют узлам, расположенным вблизи границ) 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. 70):

$$u_{ij} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}). \qquad (8.116)$$

Алгоритм решения задачи Дирихле с использованием итерационного метода решения разностных уравнений (8.116) изображен в виде блок-схемы на рис. 71. В представленном алгоритме предусмотрено вычисление 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

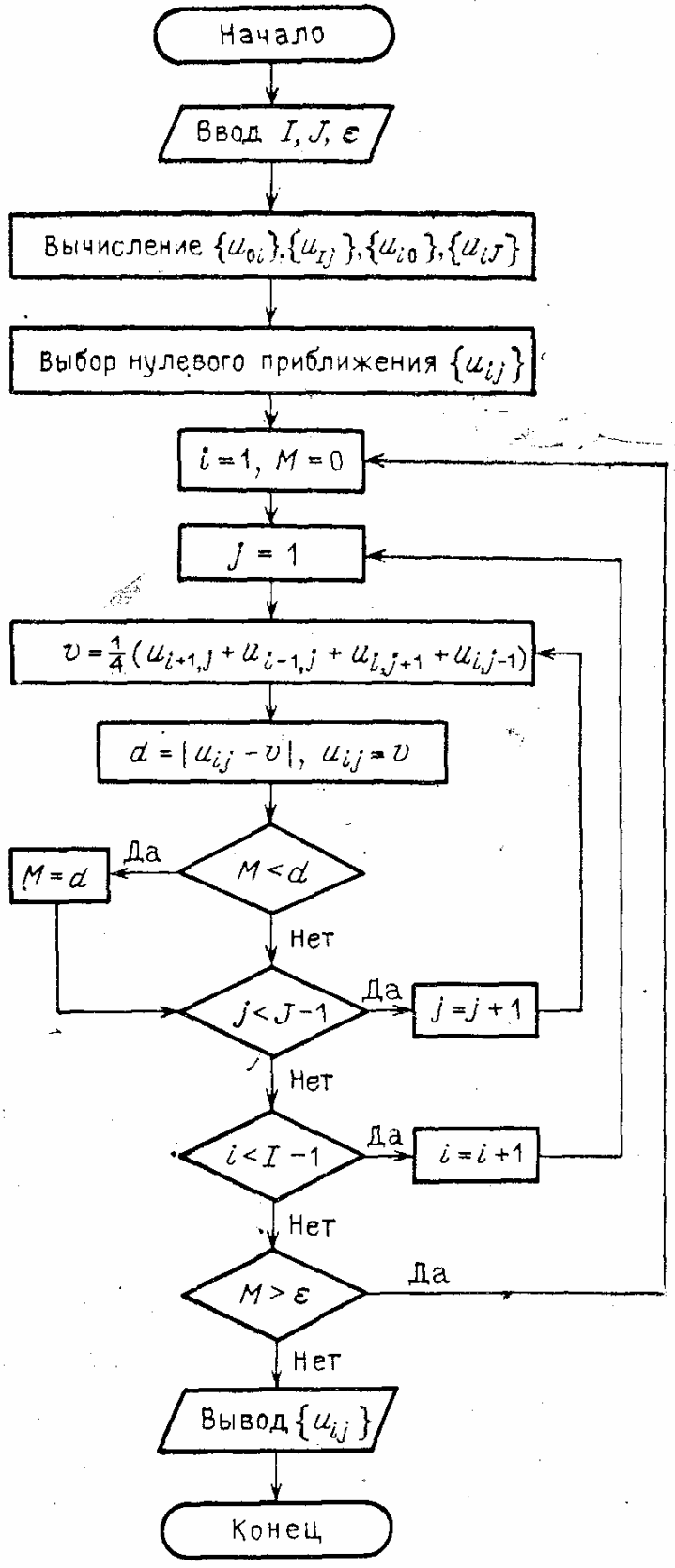


Рис. 71. Блок-схема решения задачи Дирихле

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.

Такой способ можно использовать и для решения уравнения Лапласа (8.110). Пусть требуется решить для него задачу Дирихле в прямоугольнике $ABCD$ (рис. 72).

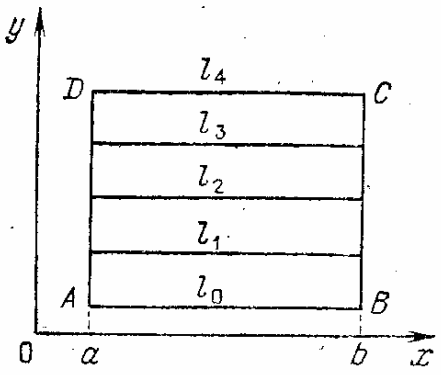


Рис. 72.

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. На каждом из этих отрезков сеточная функция u 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 сеточной функции $u_i(x)$. Аппроксимируя в уравнении (8.110) вторую производную по y с помощью отношения конечных разностей, получаем

$$\frac{d^2u_i}{dx^2} + \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = 0, \quad i = 1, 2, 3. \qquad (8.117)$$

Таким образом, решение задачи Дирихле (8.110), (8.111) сводится к решению краевой задачи для системы обыкновенных дифференциальных уравнений (8.117) относительно значений искомой функции вдоль прямых l_1, l_2, l_3. В этом состоит *метод прямых*. Граничные условия для уравнений (8.117) при $x = a$, $x = b$ можно получить из уравнений

$$u_i(a) = \varphi(a, y_i), \quad u_i(b) = \varphi(b, y_i), \quad i = 1, 2, 3. \qquad (8.118)$$

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 неза-

висимые 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

$$du_i/dx = f(u_i, t).$$

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. Построить блок-схему решения смешанной задачи для одномерного линейного уравнения переноса с использованием неявной разностной схемы.

3. Модифицировать алгоритм решения двумерного уравнения переноса (см. рис. 56) для случая, когда число слоев K не является кратным L и необходимо вывести результаты на последнем слое.

4. Построить укрупненную блок-схему нахождения разрывного решения квазилинейного уравнения переноса с использованием метода с выделением разрыва.

5. Записать алгоритм решения квазилинейного уравнения переноса по схеме сквозного счета.

6. Как изменится блок-схема решения волнового уравнения (см. рис. 65), если требуется с целью экономии памяти машины хранить не все результаты, а лишь значения сеточной функции на трех последовательных слоях?

7. Построить блок-схему решения смешанной задачи для одномерного волнового уравнения по неявной схеме.

8. Построить блок-схему решения смешанной задачи для одномерного уравнения теплопроводности: а) с помощью явной схемы; б) с помощью неявной схемы.

9. Модифицировать блок-схему решения двумерного уравнения теплопроводности (см. рис. 69) так, чтобы результаты выдавались лишь на каждом пятом слое по времени.

10. Построить блок-схему решения задачи Дирихле методом установления.

11. Записать схему расщепления по координатам для: а) двумерного волнового уравнения; б) трехмерного уравнения теплопроводности.

12. Построить блок-схему решения смешанной задачи для двумерного уравнения теплопроводности методом дробных шагов.

13. С помощью рассмотренных методов решения нестационарных задач путем ручного счета получить численные решения на первом слое при конкретных исходных данных. Для двумерного уравнения Лапласа по начальному приближению получить решение на первой итерации в узлах сетки.

ГЛАВА 9

ИНТЕГРАЛЬНЫЕ УРАВНЕНИЯ

§ 1. 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

$$\int_a^b K(x, s, y(s))\,ds = f(x, y(x)), \quad a \leqslant x \leqslant b. \qquad (9.1)$$

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

$$y_0 = y_0 + \int_{x_0}^{x} f(s, y(s))\,ds.$$

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 $x < s$, 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

$$y(x) = \lambda \int_a^b K(x, s)\, y(s)\, dx, \quad a \leqslant x \leqslant b. \tag{9.6}$$

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 $\varphi_i(x)$ — *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 $y(x) = 0$. Сле-

довательно, 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.

§ 2. Методы решения

1. Методы последовательных приближений. Это простейшие методы решения интегральных уравнений, использовавшиеся еще задолго до появления вычислительных машин.

Рассмотрим уравнение Фредгольма, записав его в виде

$$y(x) = f(x) + \lambda \int_a^b K(x, s)\, y(s)\, ds. \qquad (9.7)$$

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

$$y_{k+1}(x) = f(x) + \lambda \int_a^b K(x, s)\, y_k(s)\, ds, \quad k = 0, 1, \ldots \qquad (9.8)$$

При достаточно малом значении $|\lambda|$ и ограниченном ядре $K(x, s)$ этот итерационный процесс сходится равномерно по x, причем эта сходимость линейная. Достаточное

условие сходимости имеет вид

$$M|\lambda|(b-a)<1, \quad M=\max_{x,s}|K(x, s)|. \tag{9.9}$$

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:

$$y(x)=\sum_{k=0}^{\infty}\lambda^k\varphi_k(x). \tag{9.10}$$

Подставляя это разложение в исходное уравнение (9.7) и приравнивая выражения при одинаковых степенях λ, получаем следующие рекуррентные соотношения:

$$\varphi_0(x)=f(x), \quad \varphi_k(x)=\int_a^b K(x, s)\,\varphi_{k-1}(s)\,ds, \quad k=1, 2, \ldots \tag{9.11}$$

При ограниченных $K(x, s)$ и $f(x)$ ряд (9.10) 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:

$$K(x, s)=\sum_{i=1}^{n}\varphi_i(x)\,\psi_i(s),$$

т. е. каждый член разложения можно представить в виде произведения функций одной переменной $\varphi_i(x)$ и $\psi_i(s)$. 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 Бубнова — Галеркина. Здесь, как и при замене ядра вырожденным, для приближения решения строится аппроксимирующая система функций. Минимизация невязки аппроксимирующего уравнения проводится путем ее ортогонализации к координатным функциям.

В практических вычислениях рассмотренные методы сейчас используются сравнительно редко, поскольку присутствующие в аппроксимирующих выражениях (9.8) или (9.11) 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 ЭВМ, 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

$$\int_a^b K(x, s, y(s))\,ds = f(x, y(x)), \quad a \leqslant x \leqslant b, \qquad (9.12)$$

Разобьем отрезок $[a, b]$ на части точками $x_i = ih$ $(i = 0, 1, \ldots, n)$. Заменим интеграл в уравнении (9.12) некоторой квадратурной формулой с помощью значений сеточной функции u_i в узлах:

$$\sum_{i=1}^{n} c_i K(x_j, x_i, u_i) = f(x_j, u_j), \quad j = 1, 2, \ldots, n, \qquad (9.13)$$

где c_i — коэффициенты квадратурной формулы численного интегрирования.

Мы получили систему нелинейных алгебраических уравнений. Решая систему (9.13), 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 (9.13) для однородного уравнения Фредгольма:

$$u_j = \lambda \sum_{i=1}^{n} c_i K(x_j, x_i) u_i.$$

Запишем это выражение в виде

$$\sum_{i=1}^{n} c_i K(x_j, x_i) u_i = \frac{1}{\lambda} u_j, \quad j = 1, 2, \ldots, n. \qquad (9.14)$$

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, п. 4). Матрица A имеет n собственных значений, которые являются приближениями к собственным значениям однородного уравнения Фредгольма.

В случае неоднородного уравнения Фредгольма вместо однородной системы (9.14) получим следующую систему линейных алгебраических уравнений:

$$u_j - \lambda \sum_{i=1}^{n} c_i K(x_j, x_i) u_i = f(x_j), \quad j = 1, 2, \ldots, n. \qquad (9.15)$$

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 (9.15) становится плохо обусловленной, и сеточное решение 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 (9.15) 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 ЭВМ; системы уравнений в этих случаях более целесообразно решать итерационными методами.

Пример. Пусть задано уравнение

$$y(x) - \lambda \int_0^1 e^{-(x+s)} y(s)\, ds = x. \qquad (9.16)$$

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 $y_k = s$ $(k = 1)$ 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

$$y_3(x) = x + \lambda \int_0^1 e^{-(x+s)} y_2(s)\, ds = x + \lambda e^{-x}(a_1\lambda + a_2\lambda), \qquad (9.17)$$

$$a_1 = 1 - 2e^{-1},\ a_2 = (1 - 2e^{-1} - e^{-2} + 2e^{-3})/2.$$

В данном случае можно построить любое приближение к решению уравнения (9.16). Сходимость данного

итерационного процесса оценивается с помощью условия (9.9), которое дает ограничение на параметр λ:

$$|\lambda| < 1/[M(b-a)]. \tag{9.18}$$

Для рассматриваемого примера имеем

$$b-a=1, \quad M=\max_{x,\,s}|K(x,\,s)|=\max_{x,\,s}e^{-(x+s)}=1.$$

Следовательно, из (9.18) получаем условие $|\lambda|<1$.

Если для решения уравнения (9.16) использовать метод степенных рядов, то искомую функцию нужно представить в виде (9.10), а из рекуррентных соотношений (9.11) находим члены разложения

$$\varphi_0(x)=f(x)=x,$$

$$\varphi_1(x)=\int_0^1 K(x,\,s)\,\varphi_0(s)\,ds=\int_0^1 se^{-(x+s)}ds=e^{-x}-2e^{-(x+1)},$$

$$\varphi_2(x)=\int_0^1 K(x,\,s)\,\varphi_1(s)\,ds=\int_0^1 e^{-(x+s)}\left[e^{-s}-2e^{-(s+1)}\right]ds=$$

$$=\frac{1}{2}e^{-x}-e^{-(x+1)}-\frac{1}{2}e^{-(x+2)}+e^{-(x+3)},$$

. . . . . .

Подставляя вычисляемые значения $\varphi_i(x)$ в выражение (9.10), находим приближенное значение для искомой функции $y(x)$. Результаты совпадают с полученным ранее выражением (9.17). При $|\lambda|<1$ ряд (9.10) сходится к искомому решению.

§ 3. 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

$$\int_0^b f(x)\,dx \tag{9.19}$$

обращается в некоторой точке c отрезка $[a, b]$ в бесконечность, т. е. интеграл несобственный. Тогда его можно попытаться вычислить следующим образом:

$$\int_a^b f(x)\,dx = \lim_{\varepsilon_1 \to 0} \int_0^{c-\varepsilon_1} f(x)\,dx + \lim_{\varepsilon_2 \to 0} \int_{c+\varepsilon_2}^b f(x)\,dx. \tag{9.20}$$

Здесь ε_1, ε_2 — некоторые положительные числа, которые стремятся к нулю независимо друг от друга. Если выражения в правой части (9.20) существуют, то несобственный интеграл (9.19) сходится.

При решении ряда прикладных задач встречаются несобственные интегралы вида

$$\int_0^b \frac{dx}{x-c}, \quad c \in [a, b]. \tag{9.21}$$

В соответствии с (9.20) 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 (9.21) здесь является расходящимся.

Однако этот интеграл можно понимать и в другом смысле, полагая $\varepsilon_1 = \varepsilon_2 = \varepsilon$. В этом случае

$$\int_a^b \frac{dx}{x-c} = \lim_{\varepsilon \to 0} \left(\int_a^{c-\varepsilon} \frac{dx}{x-c} + \int_{c+\varepsilon}^b \frac{dx}{x-c} \right) =$$

$$= \lim_{\varepsilon \to 0} \left(\ln \frac{\varepsilon}{c-a} + \ln \frac{b-c}{\varepsilon} \right) = \lim_{\varepsilon \to 0} \ln \frac{b-c}{c-a} = \ln \frac{b-c}{c-a}. \tag{9.22}$$

Интеграл в таком смысле называется *интегралом в смысле главного значения по Коши* или *сингулярным интегралом*.

Аналогично можно ввести интегралы более общего вида

$$\int_a^b \frac{\gamma(x)\,dx}{x-c} = \lim_{\varepsilon\to 0}\left[\int_a^{c-\varepsilon} \frac{\gamma(x)\,dx}{x-c} + \int_{c+\varepsilon}^b \frac{\gamma(x)\,dx}{x-c}\right]. \qquad (9.23)$$

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 ряде приложений встречаются также интегралы вида

$$\int_0^{2\pi} \gamma(\theta)\operatorname{ctg}\frac{\theta-\beta}{2}\,d\theta, \quad \beta\in[0, 2\pi], \quad \gamma(\theta)\in H(\alpha). \qquad (9.24)$$

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 $x'(t)$ должна принадлежать классу H в окрестности точки $c=x(t_0)$. Например, при любом значении $c\in[-1, 1]$ имеет место тождество

$$\int_{-1}^{1} \sqrt{\frac{1-x}{1+x}}\,\frac{dx}{c-x} = \pi.$$

Это тождество можно получить, сделав в левой части дробно-линейную подстановку $t=\sqrt{(1-x)/(1+x)}$.

Тогда

$$\int_{-1}^{1} \sqrt{\frac{1-x}{1+x}} \frac{dx}{c-x} = \int_{0}^{\infty} \frac{t^2 dt}{(1+t^2)[t^2(1+c)-(1-c)]} =$$

$$= 2 \operatorname{arctg} t \Big|_0^{\infty} + \sqrt{\frac{1-c}{1+c}} \ln \left| \frac{t\sqrt{1+c} - \sqrt{1-c}}{t\sqrt{1+c} + \sqrt{1-c}} \right|_0^{\infty} = \pi.$$

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 точки $x_1, x_2, \ldots, x_n$ 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

$$\int_{-1}^{1} \frac{\gamma(x)\,dx}{x-c} \approx \sum_{k=1}^{n} \frac{\gamma(x_k)\,h}{x_k - c}.$$

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:

$$x_k = -1 + kh, \quad h = 2/(n+1), \quad k = 1, 2, \ldots, n,$$

$$c_k = x_k + h/2, \quad k = 0, 1, \ldots, n,$$

и пользуются формулой

$$\int_{-1}^{1} \frac{\gamma(x)\,dx}{x - c_m} \approx \sum_{k=1}^{n} \frac{\gamma(x_k)\,h}{x_k - c_m}, \quad m = 0, 1, \ldots, n.$$

Для интеграла с ядром Гильберта (9.24) используют следующую квадратурную формулу. Возьмем два семей-

ства точек:

$$\theta_k = k(2\pi/n), \quad \beta_k = \theta_k + \pi/n,$$
$$k = 0, 1, \ldots, n-1.$$

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:

$$\int_{-1}^{1} \frac{\gamma(x)\,dx}{x - c_m} \approx \sum_{k=1}^{n} \frac{a_k \varphi(x_k)}{x_k - c_m}, \quad m = 1, 2, \ldots, n-1,$$

$$\gamma(x) = \frac{\varphi(x)}{\sqrt{1-x^2}}, \quad \varphi^{(r)}(x) \in H(\alpha),$$

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

$$c_m = \cos \frac{m\pi}{n}, \qquad m = 1, 2, \ldots, n-1,$$

$$a_k = \pi/n, \; k = 1, 2, \ldots, n.$$

Если

$$\gamma(x) = \sqrt{1-x^2}\, \varphi(x),$$

то

$$x_k = \cos \frac{k}{n+1} \pi, \qquad k = 1, 2, \ldots, n,$$

$$c_m = \cos \frac{2m-1}{2(n+1)} \pi, \qquad m = 1, \ldots, n+1,$$

$$a_k = \frac{\pi}{n+1} \sin^2 \frac{k}{n+1} \pi, \qquad k = 1, \ldots, n.$$

Если

$$\gamma(x) = \sqrt{\frac{1-x}{1+x}}\, \varphi(x),$$

то

$$x_k = \cos\frac{2k}{2n+1}\pi, \qquad k = 1, 2, \ldots, n,$$

$$c_m = \cos\frac{2m-1}{2n+1}\pi, \qquad m = 1, 2, \ldots, n,$$

$$a_k = \frac{4\pi}{2n+1}\sin^2\frac{k}{2n+1}\pi, \quad k = 1, 2, \ldots, n.$$

2. Численное решение сингулярных интегральных уравнений. Рассмотрим следующие сингулярные интегреальные уравнения первого рода:

$$\frac{1}{\pi}\int_{-1}^{1}\frac{\gamma(x)\,dx}{x-c} + \int_{-1}^{1}K(c, x)\,\gamma(x)\,dx = f(c), \qquad (9.25)$$

$$\frac{1}{2\pi}\int_{0}^{2\pi}\gamma(\theta)\operatorname{ctg}\frac{\theta-\beta}{2}\,d\theta + \int_{0}^{2\pi}K(\beta, \theta)\,\gamma(\theta)\,d\theta = f(\beta). \qquad (9.26)$$

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 (9.25) 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 (9.25) $K(c, x) = 0$, то оно называется *характеристическим*. Его решения даются формулой

$$\gamma_\varkappa(x) = -\frac{1}{\pi}\omega_\varkappa(x)\left[\int_{-1}^{1}\frac{f(c)}{\omega_\varkappa(c)}\frac{dc}{c-x} - \nu_\varkappa C\right],$$

где $\nu_1 = 1$, $\nu_0 = \nu_{-1} = 0$, C — произвольная постоянная.

При $\varkappa = 1$ единственное решение выделяется заданием значения интеграла

$$\int_{-1}^{1} \gamma_1(x)\,dx = C.$$

При $\varkappa = -1$ функция $\gamma_{-1}(x)$ является решением характеристического уравнения только при условии

$$\int_{-1}^{1} \frac{f(x)\,dx}{\sqrt{1-x^2}} = 0.$$

В силу этого предполагают, что исходное уравнение имеет единственное решение индекса 1 при заданном значении интеграла от решения, единственное решение индекса 0 и единственное решение индекса -1 при условии

$$\int_{-1}^{1} \left[f(c) - \int_{-1}^{1} K(c, x)\,\gamma(x)\,dx \right] \frac{dc}{\sqrt{1-c^2}} = 0. \qquad (9.27)$$

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 $\varkappa = 0$:

$$\frac{1}{\pi} \sum_{k=1}^{n} \frac{\gamma_n(x_k)h}{x_k - c_m} + \sum_{k=1}^{n} K(c_m, x_k)\,\gamma_n(x_k)\,h = f(c_m),$$

$$m = 1, 2, \ldots, n-1;$$

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 (9.27). Таким образом, величина γ_{0n} в расчете является индикатором его правильности.

Если использовать неравномерное разбиение, то системы линейных алгебраических уравнений примут вид: для $\varkappa = 1$:

$$\frac{1}{\pi}\sum_{k=1}^{n}\frac{a_k\varphi_n(x_k)}{x_k - c_m} + \sum_{k=1}^{n} a_k K(c_m, x_k)\,\varphi_n(x_k) = f(c_m),$$

$$m = 1, 2, \ldots, n-1,$$

$$\sum_{k=1}^{n} a_k\varphi_n(x_k) = C,$$

$$a_k = \frac{\pi}{n}, \quad x_k = \cos\frac{2k-1}{2n}\pi, \quad c_m = \cos\frac{m\pi}{n};$$

для $\varkappa = 0$:

$$\frac{1}{\pi}\sum_{k=1}^{n}\frac{a_k\varphi_n(x_k)}{x_k - c_m} + \sum_{k=1}^{n} a_k K(c_m, x_k)\,\varphi_n(x_k) = f(c_m),$$

$$m = 1, 2, \ldots, n,$$

$$a_k = \frac{4\pi}{2n+1}\sin^2\frac{k}{2n+1}\pi, \quad x_k = \cos\frac{2k}{2n+1}\pi, \quad c_m = \cos\frac{2m-1}{2n+1}\pi;$$

для $\varkappa = -1$:

$$\gamma_{0n} + \frac{1}{\pi}\sum_{k=1}^{n}\frac{a_k\varphi_n(x_k)}{x_k - c_m} + \sum_{k=1}^{n} a_k K(c_m, x_k)\,\varphi_n(x_k) = f(c_n),$$

$$m = 1, 2, \ldots, n+1,$$

$$a_k = \frac{\pi}{n+1}\sin^2\frac{k}{n+1}\pi, \quad x_k = \cos\frac{k}{n+1}\pi, \quad c_m = \cos\frac{2m-1}{2(n+1)}\pi.$$

Для характеристического уравнения (9.26) 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:

$$\gamma_{0n} + \frac{1}{2n+1}\sum_{k=0}^{2n} \gamma_n(\theta_k)\operatorname{ctg}\frac{\theta_k-\beta_m}{2} +$$

$$+\frac{2\pi}{2n+1}\sum_{k=0}^{2n} K(\beta_m, \theta_k)\,\gamma_n(\theta_k) = f(\beta_k), \quad m = 0, 1, \ldots, 2n,$$

$$\frac{1}{2n+1}\sum_{k=0}^{2n} \gamma_n(\theta_k) = C.$$

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 (9.25), (9.26).

СПИСОК ЛИТЕРАТУРЫ

1. Аоки М. Введение в методы оптимизации: Основы и приложения нелинейного программирования/Пер. с англ.— М.: Наука, 1977.
2. Бахвалов Н. С. Численные методы.— М.: Наука, 1975.
3. Белоцерковский О. М. Численное моделирование в механике сплошных сред.— М.: Наука, 1984.
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. Воеводин В. В. Вычислительные основы линейной алгебры.— М.: Наука, 1977.
8. Волков Е. А. Численные методы.— М.: Наука, 1982.
9. Годунов С. К., Забродин А. В., Иванов М. Я., Крайко А. Н., Прокопов Г. П. Численное решение многомерных задач газовой динамики.— М.: Наука, 1976.
10. Годунов С. К., Рябенький В. С. Разностные схемы.— М.: Наука, 1977.
11. Дробышевич В. И., Дымников В. П., Ривин Г. С. Задачи по вычислительной математике.— М.: Наука, 1980.
12. Дородницын А. А. Лекции по численным методам решения уравнений вязкой жидкости.— М.: ВЦ АН СССР, 1969.
13. Дьяченко В. Ф. Основные понятия вычислительной математики.— М.: Наука, 1977.
14. Евтушенко Ю. Г. Методы решения экстремальных задач и их применение в системах оптимизации.— М.: Наука, 1982.
15. Зенкевич О. Метод конечных элементов в технике.— М.: Мир, 1975.
16. Калиткин Н. Н. Численные методы.— М.: Наука, 1978.
17. Кестенбойм Х. С., Росляков Г. С., Чудов Л. А. Точечный взрыв: Методы расчета. Таблицы.— М.: Наука, 1974.
18. Карманов В. Г. Математическое программирование.— М.: Наука, 1986.
19. Ковеня В. М., Яненко Н. Н. Методы расщепления в задачах газовой динамики.— Новосибирск: Наука, 1981.
20. Крылов В. И., Бобков В. В., Монастырный П. И. Вычислительные методы.— Т. 1. 2.— М.: Наука, 1976—1977.
21. Ляшко И. И., Макаров В. Л., Скоробогатько А. А. Методы вычислений.— Киев: Высшая школа, 1977.

22. Мак-Кракен Д., Дорн У. Численные методы и программирование на фортране.— М.: Мир, 1977.
23. Марчук Г. И. Математические модели в иммунологии.— М.: Наука, 1985.
24. Марчук Г. И. Математическое моделирование в проблеме окружающей среды.— М.: Наука, 1982.
25. Марчук Г. И. Методы вычислительной математики.— М.: Наука, 1980.
26. Марчук Г. И. Численные методы в прогнозе погоды.— Л.: Гидрометеоиздат, 1967.
27. Марчук Г. И., Лебедев В. И. Численные методы в теории переноса нейтронов.— М.: Атомиздат, 1971.
28. Михлин С. Г. Численная реализация вариационных методов.— М.: Наука, 1968.
29. На Ц. Вычислительные методы решения прикладных граничных задач.— М.: Мир, 1982.
30. Никольский С. М. Квадратурные формулы.— М.: Наука, 1979.
31. Одэн Дж. Конечные элементы в нелинейной механике сплошных сред.— М.: Мир, 1976.
32. Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем со многими неизвестными.— М.: Мир, 1975.
33. Победря Б. Е. Численные методы в теории упругости и пластичности.— М.: Изд-во МГУ, 1981.
34. Пустыльник Е. И. Статистические методы анализа и обработки наблюдений.— М.: Наука, 1968.
35. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах.— М.: Наука, 1975.
36. Рихтмайер Р., Мортон К. Разностные методы решения краевых задач.— М.: Мир, 1972.
37. Рождественский Б. Л., Яненко Н. Н. Системы квазилинейных уравнений и их приложения к газовой динамике.— Наука, 1978.
38. Роуч П. Вычислительная гидродинамика.— М.: Мир, 1980.
39. Рябенький В. С., Филиппов А. Ф. Об устойчивости разностных уравнений.— М.: Гостехиздат, 1956.
40. Самарский А. А. Введение в численные методы.— М.: Наука, 1982.
41. Самарский А. А. Теория разностных схем.— М.: Наука, 1983.
42. Самарский А. А., Гулин А. В. Устойчивость разностных схем.— М.: Наука, 1973.
43. Самарский А. А., Николаев Е. С. Методы решения сеточных уравнений.— М.: Наука, 1978.
44. Самарский А. А., Попов Ю. П. Разностные схемы газовой динамики.— М.: Наука, 1980.
45. Сегерлинд Л. Применение метода конечных элементов.— М.: Мир, 1979.

46. Соболь И. М. Численные методы Монте-Карло.— М.: Наука, 1973.
47. Стечкин С. Б., Субботин Ю. Н. Сплайны в вычислительной математике.— М.: Наука, 1976.
48. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач.— М.: Наука, 1986.
49. Уилкинсон Дж. Алгебраическая проблема собственных значений.— М.: Наука, 1970.
50. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры.— М.: Физматгиз, 1963.
51. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений.— М.: Мир, 1980.
52. Хемминг Р. В. Численные методы. Для научных работников и инженеров.— М.: Наука, 1968.
53. Худсон Д. Статистика для физиков.— М.: Мир, 1970.
54. Чушкин П. И. Метод характеристик для пространственных сверхзвуковых течений.— М.: ВЦ АН СССР, 1968.
55. Шуп Т. Решение инженерных задач на ЭВМ.— М.: Мир, 1982.
56. Яненко Н. Н. Метод дробных шагов решения многомерных задач математической физики.— Новосибирск: Наука, 1967.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Леонид Иванович Турчак

ОСНОВЫ ЧИСЛЕННЫХ МЕТОДОВ

Редактор *И. В. Викторенкова*
Художественный редактор *Т. Н. Кольченко*
Технический редактор *Л. В. Лихачева*
Корректор *М. Л. Медведская*

ИБ № 12715

Сдано в набор 14.05.86. Подписано к печати 04.12.86. Т-29452. Формат 84×108$^1/_{32}$. Бумага тип. № 3. Гарнитура обыкновенная. Печать высокая. Усл. печ. л. 16,8. Усл. кр.-отт. 16,8. Уч.-изд. л. 17,24. Тираж 43 000 экз. Заказ № 205. Цена 90 коп.

Ордена Трудового Красного Знамени
издательство «Наука»
Главная редакция физико-математической литературы
117071 Москва В-71, Ленинский проспект, 15

4-я типография издательства «Наука»
630077, г. Новосибирск-77, Станиславского, 25

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
117071 Москва В-71, Ленинский проспект, 15

ГОТОВИТСЯ К ПЕЧАТИ:

БАХВАЛОВ Н. С., ЖИДКОВ Н. П., КОБЕЛЬКОВ Г. М. Численные методы: Учеб. пособие. (Темплан 1987 г., № 54.)

Написана на основе программы курса численных методов для математических специальностей университетов. Программа отработана в содружестве педагогических коллективов Московского и Братиславского университетов (ЧССР).

Методически последовательное изложение численных методов решения задач на ЭВМ содержит строгое теоретическое обоснование. Наряду с этим четко выделена прикладная направленность методов для решения классов задач математической физики и механики. Особое внимание уделено задачам оптимизации.

Изложение ведется по схеме: описание модели математической задачи, требующей численного решения; разбор простейшего из целесообразных методов решения задачи; краткий обзор характеристик более сложных методов; рекомендации по применению методов; тестирование и отладка программ.

Для студентов математических факультетов университетов, факультетов прикладной математики, а также для аспирантов и научных работников, связанных с решением задач на ЭВМ.