М. А. ТЫНКЕВИЧ
А. Г. ПИМОНОВ

ВВЕДЕНИЕ В ЧИСЛЕННЫЙ АНАЛИЗ

Учебное пособие

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет
имени Т. Ф. Горбачева»

М. А. ТЫНКЕВИЧ А. Г. ПИМОНОВ

ВВЕДЕНИЕ В ЧИСЛЕННЫЙ АНАЛИЗ

Учебное пособие

Кемерово 2017

УДК 519.6(075.8)

РЕЦЕНЗЕНТЫ

А. М. Гудов, доктор технических наук, директор Института фундаментальных наук федерального государственного бюджетного образовательного учреждения высшего образования «Кемеровский государственный университет»

Кафедра вычислительной математики и компьютерного моделирования федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Томский государственный университет» (заведующий кафедрой А. В. Старченко, доктор физико-математических наук, профессор)

Тынкевич, М. А.

Введение в численный анализ : учеб. пособие / М. А. Тынкевич, А. Г. Пимонов ; КузГТУ. – Кемерово, 2017. – 176 с.

ISBN 978-5-906969-35-4

Данное пособие разработано на базе курса лекций по методам вычислительной математики, читавшегося в течение многих лет как для математиков-прикладников, так и для будущих специалистов в области прикладной информатики. Учитывая специфику подготовки в системе бакалавриата, авторы стремились к изложению, в первую очередь, идей основных методов численного анализа с приемлемой математической строгостью. Определенное место отведено возможностям компьютерной реализации излагаемых методов в среде MatLab.

Предназначено для обучающихся по направлениям подготовки 09.03.03 и 09.04.03 «Прикладная информатика», изучающих дисциплины «Численные методы» и «Математические и инструментальные методы поддержки принятия решений». Может быть полезно специалистам в процессе работы над проблематикой численного анализа и моделирования в разнообразных приложениях.

УДК 519.6(075.8)

ISBN 978-5-906969-35-4

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Современный представитель так называемого цивилизованного мира от Патагонии до Чукотки вопрос «Умеете ли Вы считать?» сочтет за оскорбление своего достоинства и ответит утвердительно.

Конечно, известная фраза Леопольда Кронекера «бог создал натуральные числа, а все прочее – дело рук человеческих» чревата некоторым преувеличением. Человек несколько тысяч лет назад осознал абстрактное понятие «*числа*» как меры количества каких-то объектов окружающего мира и пользовался натуральными (*nature*, англ. – *природа*) числами для исчисления барашков в стаде или потребностей княжеской дружины. В глубине тысячелетий теряется момент, когда были осознаны действия сложения и вычитания. Что касается умножения и деления, то могущественный ассирийский царь Ашшурбанипал (668−631 до н. э.) уже писал, что он «решал сложные задачи с умножением и делением, которые не сразу понятны..»*.

Затруднения возникли, когда потребовались оценки больших величин (даже по современным меркам) типа численности армии Чингиз-хана, участников крестовых походов и количества съеденных ими парнокопытных. Даже в арии индийского гостя из оперы «Садко» звучало признание: «Не счесть алмазов в каменных пещерах / Не счесть жемчужин в море полуденном…».

Наступило время, когда пришлось прибегнуть к дробным величинам – числам, представимым в виде отношения двух натуральных чисел. Так оценка отношения длины полуокружности к радиусу прошла через значения 22/7 (у Архимеда), 377/120 и т. д. Человек вынужден был научиться сложению и вычитанию обыкновенных дробей.

Прошли столетия с тех пор как человек, наблюдая за звездами, оценивая щедрость фараона или количество амфор с зерном в амбаре, пришел к азам того, что сегодня мы называем *прикладной математикой*. Древние шумеры и индусы, греки (Аристотель, Евклид, Птоломей, школа Пифагора) пришли к понятиям,

---

* Церен Э. Библейские холмы. − М. : Наука, 1966.

без знания которых в характеристике *homo sapiens* для современного человека второе слово излишне. Разумеется, эти первые шаги были сделаны немногими, избранными мыслителями в мире невежества. Говорят, что Архимед, открыв известный закон, принес в жертву богам 100 быков, и с тех пор скоты боятся всего нового. Тем не менее, незаурядная математика Древней Греции (Евклид, Пифагор, Диафант, Герон и другие) послужила фундаментом математической науки Европы в будущем. Древний Рим не был склонен к абстрактному мышлению пифагорейцев, сведя математику к землемерию и торговым расчетам.

В 41 г. до н. э. горела уникальная Александрийская библиотека при осаде Цезарем, а в 392 г. фанатиками были уничтожены ее останки. Спустя 20 лет они уничтожили последнюю из европейских математических школ (школу Гипатии Александрийской), а в 529 г. император вообще запретил «языческое обучение». Наконец, в кодексе Юстиниана (середина VII века) в разделе «Относительно математиков, звездочетов, предсказателей и подобных им злодеев» объявлялось, что «совершенно воспрещается достойное осуждения искусство математической дивинации».

Понадобилось 700 лет до Возрождения математического знания, когда Леонардо Пизанский – Фибоначчи (1170–1250) принес в Европу позиционную десятичную систему представления числовых величин. Именно она привела к удобному представлению дробных величин в режимах фиксированной точки (например, 3.14156) или экспоненциальной формы ($0.12345 \cdot 10^5$) и прогрессу в механизации и автоматизации вычислений. И лишь в конце XVI века математика вышла из стен монастырей и контор флорентийских банкиров, начав победное шествие, вторгаясь в физику, астрономию, биологию, лингвистику, экономику…

Математика редко пользовалась симпатиями властителей и необразованного окружения. Известный кардинал Ришелье в 1634 г. писал: «Науки служат одним из величайших украшений государства и обойтись без них нельзя…», но добавлял, что «усиленное знание повредит торговле и земледелию, внесет опустошение в рядах солдат, которым приличнее грубое невежество, чем тонкость знания». И спустя 200 лет император Николай I ска-

зал, что ему нужны «не умные, а верноподданные». Долгие годы немногие представители этой науки были востребованы лишь милостью филантропов подобно великому Л. Эйлеру, призванному составлять гороскопы для Анны Иоанновны (как говорят, «не на всех тронах сидят Соломоны»).

Примечательно, что в России волею Петра Великого, искавшего в Европе новых знаний для своей страны, не считавшего зазорным общаться с «худородными» Ньютоном, Лейбницем или голландскими шкиперами, видевшего в прикладной математике основу для кораблестроения и навигации, артиллерии и гидротехники, несмотря на яростное сопротивление, возникли Академия, *навигацкая* и другие *цифирные* школы, куда допускались даже «кухаркины дети» и «инородцы». Но прошли еще два столетия дворянских «недорослей», презрения к мужику и поголовной неграмотности простолюдинов.

Несмотря на черепашьи темпы распространения массового образования в стране, ускорившиеся во второй половине XIX столетия в связи с уроками Крымской войны и развитием промышленного производства, большая математика к началу XX века добралась даже до Сибири – в 1900 году открыт Томский технологический институт, а в 1917 году – физико-математический факультет Томского университета (город стали называть Сибирскими Афинами).

Сегодня стало привычным «поверять гармонию алгеброй». Создание новых типов бытовых нагревательных приборов и космических скафандров, проектирование гидростанций и линий электропередач невозможно без решения уравнений в частных производных. Разработка любой электротехнической системы, системы массового обслуживания или обеспечение условий стабильности некоторой экосистемы, в которой «и волки сыты, и овцы целы», требует решения системы обыкновенных дифференциальных уравнений. Директор универсама при установке кассовых аппаратов, стремящийся не допускать очередей у касс (покупатель может уйти к конкуренту) и не платить зарплату скучающим кассирам, вынужден решать трансцендентные уравнения. Такие уравнения возникают, как только «в бизнесе» обнаруживается нелинейная (непропорциональная) связь между ценами, объемами,

спросом и т. п. и желание найти баланс в их противоречии. Планируя объемы выпуска нескольких видов мясопродуктов при ограниченных объемах исходных ингредиентов, приходится решать системы линейных алгебраических уравнений. Банковский работник, не страдающий избытком самомнения, в попытках минимизации риска неминуемо прибегнет к методам аппроксимации данных.

Грамотная математическая постановка задачи подчас гарантирует до 75 % успеха. Однако существует много задач, для которых авторы дают различные математические постановки без доведения их решения до численного результата, который мог бы подтвердить или опровергнуть предлагаемую математическую модель. Так в экономической науке предлагается множество математических моделей, абсолютно бесполезных для реального планирования и управления.

Еще полвека назад многие задачи практически оставались неразрешимыми из-за ограниченных возможностей вычислительной техники (невысокое быстродействие, ограниченная емкость оперативной памяти, далекий от совершенства интерфейс). С тех пор вычислительная машина, занимавшая по площади десятки квадратных метров, преобразовалась в настольный компьютер, единица измерения емкости памяти «килобайты» сменилась на «гигабайты», «терабайты» и даже «петабайты», быстродействие выросло в сотни тысяч раз. Большинство современных, прагматически настроенных программистов, забыв о *программировании как искусстве*, уже не считают нужным экономить микросекунды и «ячейки памяти». Тихо скончались многие хитроумные методы, ориентированные на ручные вычисления и требующие от вычислителя незаурядной математической подготовки. Многие задачи, на решение которых великий математик Карл Гаусс тратил месяцы интенсивного труда, решаются за минуты. Использование итерационных методов (последовательных приближений), чреватое многочасовым ожиданием ответа, теперь стало обыденностью, подчас заложенной в библиотеки стандартных программ систем не только чисто математического назначения, но и даже общеизвестных электронных таблиц типа MS Excel. Методы статистического моделирования (Монте-Карло), идеи

которых связаны с эпохой Б. Паскаля и И. Бернулли, фактически родились вместе с ЭВМ и, первоначально с блеском использованные для вычисления интегралов высокой кратности, вошли в практику моделирования обыденных технологических процессов. В результате многолетней работы советских и европейских математиков-вычислителей и американских компьютерных фирм появилось достаточно много специализированных систем для решения математических задач с богатейшими библиотеками процедур (MatLab, MatCad, Mathematica, Maple, Maxima, SciLab и другие).

Сегодня человек, получивший начальное образование (предел мечтаний миллионов жителей дореволюционной России) и, тем более, вооруженный калькулятором или компьютером, в рамках обыденной деятельности уверен в своих вычислительных способностях. И он абсолютно прав, если его математические потребности ограничены необходимостью лишь сопоставлять доходы и расходы в его небольшом хозяйстве. Прочитав на телеэкране, что в России 146 544 710 жителей или что эффективность приема неких чудодейственных пилюль составляет 99.567 %, он спокойно проглотит такую фантастическую точность? А подписывая смету расходов на ремонт котельной в объеме 245045.72 руб.? На экране телевизора мелькают изящные, гладкие, непрерывные кривые роста благосостояния, колебаний цен на нефть или рейтинга претендентов в избирательной гонке. Откуда они берутся? Ведь не каждую секунду производятся замеры.

Если ваш коллега решает уравнение $\mathrm{tg}\, x^2 = 1$ в виде $x = 1/\sqrt{\mathrm{tg}}$, его диплом о высшем образовании, скорее всего, куплен на одном из рынков.

При поиске решения задач прикладной математики, возникающих в той или иной сфере деятельности, необходим учет следующих истин.

Во-первых, непосвященный может оказаться в положении «буриданова осла», не сумев найти среди предлагаемых стандартных процедур приемлемую для решения поставленной задачи. Так, решая систему линейных алгебраических уравнений,

из десятков предлагаемых численных процедур он, наверняка, выберет знакомый с детства метод подстановок (метод Гаусса), получая подчас результат, далекий от истины.

Во-вторых, надежда на легкое решение задач использованием стандартных процедур подчас обманчива. Так попытка использовать таковые для задач «солидной» размерности часто заканчивается сообщением: «точность не достигнута в результате … итераций».

В-третьих, большинство исследовательских задач заставляет, не обнаружив готовых программных средств, создавать оригинальные процедуры на основе уже известных подходов. Но сколько раз авторитетные специалисты, заменяя систему дифференциальных уравнений конечно-разностным аналогом без дополнительного анализа на существование и единственность решения, получали нечто фантастическое и в соседнем городке регулярно вылетали стекла в жилых домах.

Существующие стандартные средства значительно облегчают решение задач, но иллюзорна возможность обойтись без знания методов вычислений. Можно, конечно, встать на позицию известного в свое время недоросля Митрофанушки Простакова, высказанную им относительно нужды в географических познаниях, но и сегодня нужны знающие географию «извозчики».

Существует обширная литература по методам численного анализа, причем ее «звездные часы» связаны с серединой XX века, когда вычислительная техника только зарождалась, где изложена идеология и дано глубокое обоснование основных методов численного анализа, требующее от читателя фундаментальной математической подготовки. Первые программисты той эпохи были вынуждены при разработке стандартных программ экономить каждую микросекунду и каждую ячейку памяти ЭВМ, тем самым создавая «шедевры искусства программирования». В последние годы появилось множество учебников, отказывающихся от изложения оригинальных методов, которые еще 10 лет назад были в работе, и учитывающих фантастический рост быстродействия и памяти современных компьютеров. Многое можно обнаружить в среде Интернета, если иметь представление о базовых

вычислительных методах и не доверять слепо содержащейся там массе ошибок и нелепостей.

Мы не пытаемся «объять необъятное», хотя каждому способному читать и понимать умные книги к этому следует стремиться, познавать (может быть, и творить) новое в будущей деятельности. За бортом предлагаемого руководства остаются многие методы численного анализа, связанные с решением уравнений математической физики и специальными разделами высшей математики. Авторами предпринята попытка сочетать необходимую математическую строгость изложения, способность восприятия этих истин читателем, не обладающим основательной математической подготовкой, и подсказать возможность компьютерной реализации излагаемых методов, в частности, в среде MatLab. Мы излагаем идеи лишь базовых методов, без знакомства с которыми не существует грамотного специалиста – инженера – исследователя, пытаясь создать фундамент для последующего самообразования за счет доступа к фундаментальным литературным источникам и разумному поиску в необъятном мире Интернета.

Глава 1. ДЕЙСТВИЯ НАД ПРИБЛИЖЕННЫМИ ВЕЛИЧИНАМИ

Выше мы уже цитировали известного математика Л. Кронекера (1823–1891), заявлявшего: «die ganze Zahlen hat der liebe Gott gemacht, alles anderes ist Menshenwerk» (целые числа созданы богом, все остальное – творение человека).

Прежде чем окунуться в мир приближенных вычислений, с которыми мы имеем дело с момента знакомства с десятичными дробями, рассмотрим элементарную задачу [3].

В кожух квадратного сечения (со стороной $2R$) помещен цилиндр радиуса R (рис. 1.1). По углам кожуха размещены сферические катки неизвестного радиуса r. Требуется определить объем таких катков. Решая эту задачу, из элементарного соотношения $(R + r)^2 = 2\,(R - r)^2$ получаем значения радиуса r и, соответственно, искомого объема:

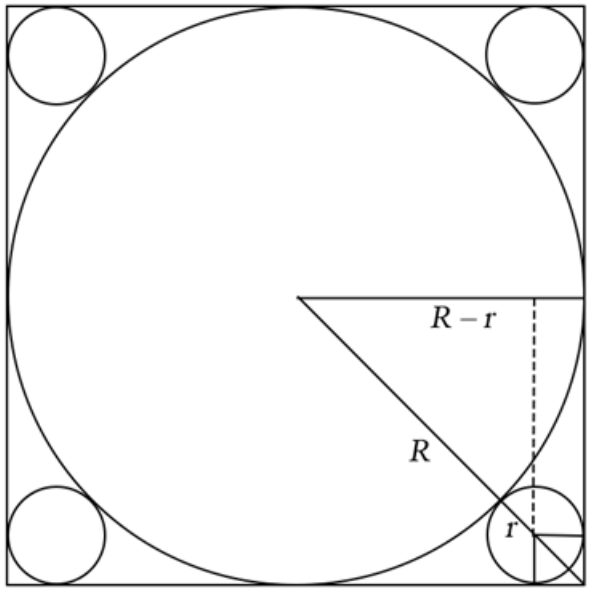


Рис. 1.1

$$V = \frac{4}{3}\pi R^3\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)^3.$$

Очевидно, что фигурирующий здесь куб отношения можно вычислить различными путями:

$$\left(\frac{\sqrt{2}-1}{\sqrt{2}+1}\right)^3 = \left(\sqrt{2}-1\right)^6 = (3-2\sqrt{2})^3 = 99-70\sqrt{2}.$$

Если под рукой нет калькулятора и единственными техническими средствами являются лист бумаги и авторучка, избегая многозначных чисел, примем значение $\sqrt{2}$ равным 1.4. Проведя соответствующие вычисления, обнаруживаем парадокс – приведенные выше формулы дают существенно разные значения 0.00463, 0.004096, 0.008 и 1 (истинное же значение, округленное до 7 знаков после десятичной точки, равно 0.0050506). Вот вам

пример того, как небольшая погрешность в исходных данных может привести к неверным итогам.

Конечно, выбор более точной оценки $\sqrt{2}$ даст более правдоподобный результат, если не будем злоупотреблять округлениями в процессе вычислений, но объем вычислительной работы существенно возрастет (при ручных вычислениях). Вычисления с помощью компьютеров или калькуляторов обеспечивают получение ответа в форме 6–10-значных чисел, но нет гарантии того, что все цифры полученного ответа верны.

Наш замечательный кораблестроитель, механик и математик академик А. Н. Крылов* говорил, что ему приходилось рассматривать проекты, в которых 90 % работы затрачивалось впустую на выписывание ненужных и неверных цифр. Были случаи, когда налоговые службы требовали оплатить 0 руб. 00 коп., если в компьютерной программе не было предусмотрено округление расчетных значений до копеек. Забавно читать, что Г. Потемкин, одобрив план создания города Екатеринослав в Новороссии, в 1776 г. утвердил смету расходов в 137149 руб. и 32.5 коп. Но до сих пор в некоторых публикациях приходится сталкиваться с оценкой длины беговой дорожки 287.02345 м, фонда заработной платы 8 256 023.0567 руб. или эффективности нового лекарства для похудения 78 %.

Возникают естественные вопросы. Какова точность полученного результата? С какой точностью нужно задавать исходные показатели для получения результата с заданным числом верных знаков (цифр)? Повлияют ли порядок вычислений или выбор метода на итог расчета?

Все вычислительные погрешности можно разделить на три группы.

1. Первая из них связана с *погрешностью округлений* в процессе вычислений. От нее практически невозможно избавиться; даже при компьютерном счете она не исчезает, хотя за

---

* Алексей Николаевич Крылов (1863–1945) – выдающийся специалист в области математики и механики, знаменитый кораблестроитель, вместе с адмиралом С. О. Макаровым создавший теорию непотопляемости, автор превосходного учебника по приближенным вычислениям.

счет представления в формате чисел с двойной точно*стью* (`double precision`) и режима плавающей точки (`float point`) она уменьшается до незначимых величин.

2. При задании исходных данных мы, как правило, берем не истинные оценки, а приближенные. Объявляемый продавцом вес товара зависит от того, под каким углом он видит стрелку весов. Заработная плата рабочего выступает как приближенная оценка истинных затрат его труда. Большинство физических констант найдено в результате эксперимента. Использование подобных величин приводит к *неустранимой погрешности*, или *погрешности исходных данных*.

3. Одну и ту же задачу можно решать разными методами, каждый из которых вносит в результат свою погрешность – *погрешность метода*.

1.1. Абсолютная и относительная погрешности

Пусть α – приближенное значение некоторого числа A, точного значения которого мы не знаем. *По возможности малое число* $\Delta > 0$ такое, что $|\alpha - A| < \Delta$, называют *предельной абсолютной погрешностью* (слово «предельный» обычно опускают). Если в истинном значении $\sqrt{2} = 1.4142135623\ldots$ выполнить общепринятое округление до двух знаков после десятичной точки $\sqrt{2} \approx 1.41$, то абсолютная погрешность такого представления не превышает 0.005.

Абсолютная погрешность не всегда дает полную характеристику результата вычислений. Так при оценках, связанных с миллиардами рублей, точность до рубля едва ли разумна и недостижима (прогноз прибыли Сбербанка РФ на следующий год с такой точностью вызывает в лучшем случае усмешку читателя). С другой стороны, та же точность при оценке затрат на изготовление экземпляра разменной монеты едва ли приемлема. Абсолютная погрешность в 1 мм ничтожна при оценке расстояния от Москвы до Рио-де-Жанейро и абсурдна для расстояний между молекулами твердого вещества. Поэтому часто используется понятие *предельной относительной*

погрешности $\delta = \dfrac{\Delta}{|A|}$ (очевидно, что в реальности при неизвестном A приходится использовать деление не на |A|, а на |α|).

Элементарный здравый смысл подсказывает, что при работе с малыми (например, существенно меньшими 1) или большими (например, значимо превышающими 10) величинами более содержательную информацию несет относительная погрешность. В согласии с этим построено представление числовой информации в памяти компьютеров.

1.2. Компьютерное представление числовых величин

Есть ли резон говорить о вычислительной погрешности, если ваш калькулятор дает ответ с 15 десятичными знаками? Пока мы имеем дело с оценками, не выходящими за пределы обыденности, погрешность наших округлений ничтожна и о ней можно забыть, поскольку даже дробь 2 / 3 здесь равна 0.66666…667.

Напомним некоторые общеизвестные ныне истины.

Читатель, знакомый с компьютером хотя бы в объеме средней школы, наверное, слышал, что всякое число можно представить в порядковой системе счисления с тем или иным основанием q (обычно q равно 10 при ручных вычислениях и 2 при внутрикомпьютерных представлениях, хотя на заре появления ЭВМ существовали машины и на базе троичной системы)

$$a = \pm(a_1 q^n + a_2 q^{n-1} + ... + a_m q^{n-m+1} + ...),$$

где a_i – целые числа из диапазона от 0 до $q - 1$. Например:

$$17.50 = 1 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 0 \cdot 10^{-2} + ...;$$

то же число в шестнадцатиричной системе

$$17.50 = 1 \cdot 16^1 + 1 \cdot 16^0 + 8 \cdot 16^{-1} + 0 \cdot 16^{-2} + ... = 11.80_{16}$$

и в двоичной

$$17.50 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + ... = 10001.1000_2.$$

При представлении чисел в памяти компьютеров используются две основные формы записи – с фиксированной (`fixed point`) или плавающей (`float point`) точкой.

В режиме `fixed point` обычно один двоичный разряд отводится на запись знака числа и остальные – на изображение его абсолютной величины (положение разделительной точки определяется устанавливаемым форматом). В однобайтовом поле (8 разрядов – бит) в зависимости от уговора о положении точки (считать число дробным, меньшим единицы, или целым) можно изображать числа из диапазона от 2^{-7} до $2^8 - 1 = 127$, в поле двойного слова (64 разряда) − от 2^{-63} до $2^{64} - 1$. Попытка записи чисел, меньших 2^{-7} или 2^{-63}, дает *машинный нуль*.

В режиме `float point` используют так называемую *нормальную форму*

$$a = \pm x \cdot q^p,$$

где $1 / q \le |x| < 1$ – *мантисса* и p – *порядок* числа (например, $17.5 = 0.175 \cdot 10^2$, $0.000057 = 0.57 \cdot 10^{-4}$). В компьютерной записи один байт используется для отображения знака числа (1 бит), знака порядка (1 бит) и абсолютной величины порядка (6 бит). Мантисса записывается в остальных 3 или 7 байтах поля («слова» или «двойного слова»). Поэтому для поля типа «слово» (формат `real` в ряде систем программирования) диапазон допустимых значений составляет $[0.5 \cdot 2^{-64}, (1 - 2^{-24}) \cdot 2^{63}]$, что соответствует значениям от 10^{-20} до 10^{19} и 10–11 *значащим* десятичным *цифрам*. Выход за максимально возможный предел обычно ведет к прерыванию работы компьютера с сообщением о переполнении разрядной сетки (`overflow`).

Объявляя режим плавающей точки, мы автоматически приходим к приближенному представлению чисел, дробная часть которых не является конечной суммой степеней 2. Так двоичная запись мантиссы числа $0.1 = 0.8 \cdot 2^{-3}$ в 3×8 двоичных разрядах имеет вид 0.1100 1100 1100 1100 1100 110**1** (правило округления), т. е. машинная запись числа превышает истинное его значение на величину порядка $(2^{-25}) \cdot 2^{-3} \approx 10^{-28}$. Вроде бы погрешность ничтожна, но при компьютерном расчете, прибавив к нулю 10 раз двоичный эквивалент 0.1, получим значение, большее 1. Цикл с заголовком типа `while x≠1 do x:=x+0.1` не имеет шансов на завершение.

1.3. Погрешности элементарных операций

При сложении и вычитании абсолютная погрешность результата не превышает суммы абсолютных погрешностей операндов

$$\Delta_{a+b} = |a + b - (A + B)| = |(A \pm \Delta_a) + (B \pm \Delta_b) - (A + B)| \le \Delta_a + \Delta_b.$$

При ручном счете, суммируя операнды, имеющие разную абсолютную погрешность, выбирают операнд с максимальной погрешностью и остальные округляют с сохранением лишнего знака. Так при поиске 3.1 + 65.626 + 2.76435 достаточно найти сумму 3.1 + 65.63 + 2.76 =70.49 с последующим округлением до 70.5.

Для абсолютной погрешности суммы значительного количества n слагаемых приемлема менее завышенная *вероятностная* оценка

$$\Delta = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \Delta x_i ,$$

где Δx_i – абсолютные погрешности слагаемых.

Что касается оценки относительной погрешности при суммировании, можно лишь показать, что при сложении чисел с одинаковыми знаками *относительная погрешность суммы не превышает наибольшей относительной погрешности операндов*

$$\delta\left(\sum_i x_i\right) = \Delta\left(\sum x_i\right) \Big/ \left|\sum x_i\right| < \max_i \delta\, x_i .$$

При вычитании относительная погрешность результата, близкого к нулю, может оказаться большой. Так при близких величинах a = 47.132 и b = 47.111 (относительная погрешность операндов равна 0.0005 / 47 ≈ 0.00001) относительная погрешность разности $a - b$ = 0.021 равна 0.0005 / 0.021 ≈ 0.05, т. е. возросла в 5000 раз.

Относительная погрешность произведения и частного не превышает суммы относительных погрешностей операндов

$$\delta_{ab} = \Delta_{ab} / |ab| = \left|\frac{(A+\Delta a)(B+\Delta b)-AB}{ab}\right| = \left|\frac{B\cdot\Delta a + A\cdot\Delta b + \Delta a\cdot\Delta b}{ab}\right| \approx$$

$$\approx \left|\frac{B\cdot\Delta a + A\cdot\Delta b}{ab}\right| \le \left|\frac{B}{b}\cdot\frac{\Delta a}{a}\right| + \left|\frac{A}{a}\cdot\frac{\Delta b}{b}\right| \approx \delta a + \delta b,$$

$$\delta_{a/b} = \Delta_{a/b} / |a/b| = \left|[\frac{(A+\Delta a)}{(B+\Delta b)} - \frac{A}{B}]/(a/b)\right| =$$

$$= \left|[\frac{(A+\Delta a)(B-\Delta b)}{B^2-(\Delta b)^2} - \frac{A}{B}]/(a/b)\right| = \left|[\frac{[A\cdot B - A\cdot\Delta b + B\cdot\Delta a - \Delta a\cdot\Delta b]}{B^2-(\Delta b)^2} - \frac{A}{B}]\cdot\frac{b}{a}\right| \approx$$

$$\approx \left|[-\frac{A}{B}\cdot\frac{\Delta b}{B} + \frac{\Delta a}{B}]\cdot\frac{b}{a}\right| = \left|-\frac{Ab}{aB}\cdot\frac{\Delta b}{B} + \frac{b\cdot\Delta a}{aB}\right| \approx \left|-\frac{\Delta b}{B} + \frac{\Delta a}{a}\right| \le$$

$$\left|\frac{B}{b}\cdot\frac{\Delta a}{a}\right| + \left|\frac{A}{a}\cdot\frac{\Delta b}{b}\right| \approx \delta a + \delta b,$$

абсолютная же погрешность зависит от значений самих операндов и при делении, например, на число, близкое к нулю, может оказаться большой.

1.4. Значащие цифры и верные знаки

Не прибегая к формальным определениям, заметим, что в записи 512.430 (можете еще приписать нули справа) присутствует 6 значащих цифр, а в записи 0.0023 – 2 значащие цифры. То же можно сказать и для экспоненциальных представлений $0.512430\cdot10^3$ и $0.23\cdot10^{-2}$.

Для числа 512.43… при абсолютной его погрешности 0.08 диапазон истинного значения от 512.35 до 512.51 и совпадение трех первых цифр позволяют назвать их *верными*. Остальные цифры *сомнительны*.

Записав значение 3.1415, мы гарантируем верность пяти значащим цифрам и абсолютную погрешность 0.00005. Для простоты оценок *относительная погрешность принимается равной абсолютной погрешности, деленной на удвоенную первую его значащую цифру*. Для нашего примера относительная погрешность равна 0.00005/6.

Еще раз обратите внимание, что записи 1 млн. и 1 000 000 руб. не эквивалетны – их абсолютные погрешности не превышают 0.5 млн. рублей и 50 коп. Получив оценку 123456 с гарантией лишь 4 верных знаков, мы должны записать ее в виде $1234 \cdot 10^2$, $0.1234 \cdot 10^6$ и т. п.

1.5. Вычисление значений функций и формула Тейлора

Прежде чем вести разговор о погрешностях, возникающих при вычислении значений выражений, содержащих не только четыре арифметические операции, обратимся к проблемам вычисления значений элементарных аналитически заданных функций.

1.5.1. Вычисление значений алгебраического многочлена

Попытаемся найти значение популярной элементарной функции – многочлена n-й степени

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

при конкретном значении x.

Вооружившись какой-либо программной средой, реализуем эти вычисления записью одного оператора присваивания или циклом, в котором используется заданный массив значений `A` с индексами от `0` до `n`:

```
P:=0; for k:=0 to n do P:=P+A[k]*x^(n-k);
```

Вычисление целочисленных степеней x сводится к умножениям и оба варианта потребуют $n + (n - 1) + (n - 2) + \ldots + 2 + 1 = n(n + 1) / 2$ умножений.

Перепишем многочлен с использованием так называемых *скобок* (*схемы*) *Горнера*[*]

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

Записав алгоритм вида

```
P:=A[0]; for k:=1 to n do P:=P*x+A[k];
```

обнаруживаем, что потребуется лишь n умножений.

---

* Вильямс Джордж Горнер (1786–1837) – английский математик. Предложил способ приближенного вычисления вещественных корней полинома, схему деления многочлена на двучлен и упомянутый способ вычисления значений полинома.

Кстати, схема Горнера лежит в основе известных правил перевода десятичных чисел в другие системы счисления.

1.5.2. Формула Тейлора в случае одной переменной

Аналитическая (непрерывная дифференцируемая) функция $F(x)$ представима в виде степенного ряда – при $x = x^*$ разлагается в окрестности x^* в сходящийся ряд Тейлора*

$$F(x) = F(x^*) + F'(x^*)(x - x^*) + \frac{F''(x^*)}{2!}(x - x^*)^2 + \ldots$$

$$\ldots + \frac{F^{(n)}(x^*)}{n!}(x - x^*)^n + \ldots$$

(при $x^* = 0$ этот ряд называется рядом Маклорена**).

Представьте себе, что вам необходимо найти значение синуса 400° при отсутствии копьютера. Для функции $\sin(x)$ ряд Маклорена

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots + (-1)^{n-1}\frac{x^{2n-1}}{(2n-1)!} + \ldots$$

Естественно перевести значение аргумента из градусной меры в радианную $x = 400°/180°\cdot\pi = 2\frac{2}{9}\cdot\pi$ радиан. С учетом периодичности функции уменьшаем x на 2π, получая $x = \frac{2}{9}\pi \approx 0.636626$. Тогда $\sin(x) = 0.6981 - \frac{0.6981^3}{3!} + \frac{0.6981^5}{5!} - \ldots = 0.6981 - 0.0567 +$ $+ 0.0014 - 1.603719 \cdot 10^{-5} + \ldots = 0.6428$. Заметьте, что уже четвертый член нашего знакочередующегося ряда меньше $0.5 \cdot 10^{-4}$ и не влияет на четвертую значащую цифру.

---

* Брук Тейлор (1685–1731) – английский математик, наряду с публикацией данной формулы (1715), занимался разнообразными вопросами теории колебаний и др.

** Колин Маклорен (1698–1746) – выдающийся английский математик, известен решением различных задач геометрии, механики и астрономии.

Высказанные замечания лежат и в основе практически всех стандартных процедур вычисления различных функций*. Так для показательной функции

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

при $|x| < 1$ восьмое слагаемое равно $0.25 \cdot 10^{-4}$ и сумма восьми членов ряда гарантирует не менее пяти верных знаков.

Ряды не бесполезны и в более сложных случаях. Например:

$$\int_0^z \frac{\sin(x)}{x}\,dx = \int_0^z \frac{1}{x}\left[x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots\right]dx = 1 - \frac{z^3}{3 \cdot 3!} + \frac{z^5}{5 \cdot 5!} - \cdots$$

В этом случае имеет место ряд с весьма быстрым убыванием слагаемых (по модулю) при небольших значениях аргумента.

1.5.3. Формула Тейлора и итерационные методы

Каждый старшеклассник умеет «искать» квадратный корень и пасует перед корнями большего порядка. Обратимся к задаче поиска $y = \sqrt[k]{x}$. Заменим ее задачей решения уравнения $F(y) = y^k - x = 0$ и воспользуемся разложением $F(y)$ в ряд Тейлора в окрестности значения y_n (ограничимся учетом только первых двух ее членов):

$F(y) \cong F(y_n) + (y - y_n)\,F'(y_n) = y_n^k - x + (y - y_n)\,k\,y_n^{k-1} = 0,$

откуда получаем улучшенное приближение $y = y_{n+1}$ для искомого корня

$$y_{n+1} = \frac{k-1}{k} y_n + \frac{x}{k y_n^{k-1}}.$$

Задавшись некоторым начальным значением y_0, отыскиваем последующие приближения до тех пор, пока они не окажутся близ-

---

* Библиотеки стандартных программ вычисления значений многих функций для ЭВМ с хранением на магнитной ленте в СССР разрабатывались и пополнялись с 1956 года. В первой серийной ЭВМ «Стрела» часть такой библиотеки была даже зашита в постоянной памяти. Качеству использованного математического аппарата могут позавидовать и современные разработчики программного обеспечения – приходилось экономить каждую ячейку памяти и каждую микросекунду.

кими в смысле заданной абсолютной или относительной погрешности. Например, для кубического корня такой процесс последовательных приближений (итерационный процесс)

$$y_{n+1} = \frac{1}{3}\left(2y_n + \frac{x}{y_n^2}\right), n = 0, 1, 2, \ldots$$

при $x = 10$ и $y_0 = 2$ дает $y_1 = 2.083$, $y_2 = 2.156$, $y_3 = 2.154$ и т. д.

Однако, взяв уравнение $4x - 4 = 0$, преобразовав его к виду $x = 4 - 3x$ и запустив итерационный процесс $x_{n+1} = 4 - 3x_n$, получаем $x_0 = 0$, $x_1 = 4$, $x_2 = -8$, $x_3 = 28$, $x_3 = -80$ и т. д. Очевидно, возник расходящийся итерационный процесс, нарушены условия сходимости, обсуждение которых последует ниже.

Итерационные методы эффективны для решения многих задач, особенно задач большой размерности. Достоинством итерационных процедур является возможность получения результата с любой требуемой точностью и их устойчивость к промежуточным ошибкам (незначительные арифметические ошибки могут замедлить процесс приближений, не влияя на конечный результат). Однако при таком подходе должна быть уверенность в сходимости процесса и задан какой-то критерий для выбора начального приближения.

1.6. Прямая и обратная задачи теории погрешностей

Очевидно, что при массовых вычислениях никто не занимается утомительной пооперационной оценкой погрешностей. При ручном счете используют 1-2 *лишних* значащих цифры, что избавляет от лишней и бесполезной работы, но в итоговых оценках учитывают присутствие наименее точных исходных данных. При машинном счете стараются избежать гигантской погрешности из-за вычитания близких величин, делений на числа, близкие к нулю, умножений очень маленьких чисел на очень большие. Получив девятизначный результат, не надейтесь на правильность всех цифр. Имейте в виду − если отбрасываемая цифра 5 и за ней последуют нулевые, работает *правило четной цифры*: увеличение последней значащей выполняется лишь в случае, если она нечет-

ная. Так, округляя числа 1.2500 и 1.3500 до двух значащих цифр, получаем 1.2 и 1.4

Если воспользоваться известной формулой Тейлора для функции нескольких переменных

$$F(x_1 + \Delta x_1, x_2 + \Delta x_2, \ldots, x_n + \Delta x_n) =$$

$$= F(x_1, x_2, \ldots, x_n) + \sum_{i=1}^{n} \frac{\partial F}{\partial x_i} \Delta x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 F}{\partial x_i \, \partial x_j} \Delta x_i \, \Delta x_j + \ldots$$

и ограничиться только линейной ее частью, то для абсолютной и относительной погрешностей при вычислении значений функции $F(X)$ можно установить приближенные оценки вида

$$\Delta F = \sum_i \left| \frac{\partial F}{\partial x_i} \right| \Delta x_i, \; \delta F = \sum_i \left| \frac{\partial}{\partial x_i} \ln(F) \right| \Delta x_i, \qquad (1.1)$$

где Δx_i – абсолютные погрешности аргументов.

Так при вычислении квадратного корня обнаруживаем

$$\Delta(\sqrt{x}) = \frac{1}{2\sqrt{x}} \Delta x, \; \delta(\sqrt{x}) = \frac{d}{dx} \ln(\sqrt{x}) \, \Delta x = \frac{1}{2} \delta x,$$

а для значений $F(a, b) = a^2 + e^{a \sin(b)}$ получаем

$$\Delta F \approx \left| 2a + \sin(b) \, e^{a \sin(b)} \right| \Delta a + \left| \cos(b) \, e^{a \sin(b)} \right| \Delta b.$$

Наряду с рассмотренной *прямой задачей* оценки погрешности итога вычисления некоторого выражения при известных погрешностях его параметров определенный интерес представляет и *обратная задача теории погрешностей* – с какой погрешностью достаточно задать исходные параметры, чтобы обеспечить требуемую точность результата? Эта задача математически некорректна и может иметь множество решений. Одно из них базируется на *принципе равных влияний* (предположение, что каждый из параметров вносит одинаковую абсолютную погрешность в общую погрешность результата) и имеет вид

$$\Delta x_i = \frac{|x_i| \Delta F}{\sum_i \left| x_i \frac{\partial F}{\partial x_i} \right|}. \qquad (1.2)$$

Пример 1. Попробуем оценить погрешность результата вычисления значения $F(a, b, t) = (a^2 + b^3) / \cos(t)$, если $a = 28.3 \pm 0.02$, $b = 7.45 \pm 0.01$, $t = 0.7854 \pm 0.0001$.

Абсолютные погрешности исходных данных:

$$\Delta a = 0.02,\ \Delta b = 0.01,\ \Delta t = 0.0001.$$

Относительные погрешности исходных данных:

$$\delta a = 0.02 / 28.3 = 0.00071,\ \delta b = 0.01 / 7.45 = 0.00135,$$

$$\delta t = 0.0001/ 0.7854 = 0.00013.$$

Ориентируясь на (1.1), имеем

$$F = (a^2 + b^3) / \cos(t) = 1214.4 / 0.7071 = 1717.44,$$

$$\frac{\partial F}{\partial a} = 2a \Big/ \cos(t) = 80.05,\ \frac{\partial F}{\partial b} = 3b^2 \Big/ \cos(t) = 235.48;$$

$$\frac{\partial F}{\partial t} = -\frac{a^2 + b^3}{\cos^2 t} \sin(t) = 1717.4;$$

$$\Delta F = \left|\frac{\partial F}{\partial a}\right| \Delta a + \left|\frac{\partial F}{\partial b}\right| \Delta b + \left|\frac{\partial F}{\partial t}\right| \Delta t = 4.1275;$$

$$\delta F = \Delta F / F = 0.0024\ (\approx 0.25\ \%).$$

Оценив диапазон возможных значений величины F, обнаруживаем доверие к первым трем цифрам и, выполнив округление, имеем итог $\boldsymbol{F = 1.72 \cdot 10^3}$ (но не 1720).

Пример 2. С какой точностью нужно задать параметры a, b, t при вычислении значения $F = (a^2 + b^3) / \cos(t)$ с $m = 5$ верными знаками, если заданы $a \approx 28.3$, $b \approx 7.45$, $t \approx 0.7854$?

Находим $a^2 = 800.9$, $b^3 = 413.5$, $\cos(t) = 0.7071$, $a^2 + b^3 = 1214.4$, $F = (a^2 + b^3) / \cos(t) = 1214.4 / 0.7071 = 1717.4$ (результат записан с пятью значащими цифрами, но следует ли всем им доверять?).

Ориентируясь на (1.2), находим

$$a\ (dF / da) = 2a^2 / \cos(t) = 1601.9 / 0.7071 = 2265.45,$$

$$b\ (dF / db) = 2b^3 / \cos(t) = 827.0 / 0.7071 = 1169.57,$$

$$t\ (dF / dt) = t\ (a^2 + b^3) / \cos^2(t) \sin(t) = 0.7071 \cdot 1214.4 / 0.7071 = 1214.4,$$

знаменатель: $2265.45 + 1169.57 + 1214.4 = 4649.4$.

Отсюда допустимая погрешность исходных параметров равна

$$\Delta a = 28.3 \cdot 0.008572 / 4649.4 = 0.00005 = 0.5 \cdot 10^{-4};$$

$$\Delta b = 7.45 \cdot 0.008572 / 4649.4 = 0.00001 = 0.1 \cdot 10^{-4};$$
$$\Delta t = 0.7071 \cdot 0.008572 / 4649.4 = 0.000001 = 0.1 \cdot 10^{-5}.$$

Сравнивая найденные оценки с фактическими погрешностями входных величин ($\Delta a = 0.05$, $\Delta b = 0.005$, $\Delta t = 0.00005$), видим, что доверять пяти цифрам в найденном значении F нельзя.

1.7. Полезно вспомнить

При выполнении оценок сходимости вычислительных процессов, при вычислении точности получаемых оценок, при представлении ряда выражений в компактной форме, при решении элементарных задач комбинаторики небесполезно знать о приведенных ниже понятиях (конечно, читатель с ними знаком чуть ли не со школьной скамьи, но на всякий случай мы о них напоминаем).

Функция – правило, по которому элементам некоторого множества сопоставляются элементы другого множества. **Вычислить функцию невозможно** – можно вычислить только *значение* функции.

Факториал – функция, обозначаемая $n!$ и определенная для неотрицательных целочисленных аргументов следующим образом: $\boldsymbol{n!} = 1 \cdot 2 \cdot \ldots \cdot (n-1) \cdot n$; $0! = 1! = 1$. При больших значениях n применима формула Стирлинга $n! \approx \sqrt{2\pi n}\, n^n e^{-n}$. Так что запись **5!** едва ли связана с восторгом от отличной оценки.

Обобщением факториала для нецелого аргумента выступает известная *гамма-функция* (в библиотеке MatLab используется под именем `gamma`)

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt, \; \Gamma(n+1) = n!$$

Скалярное произведение n-мерных векторов $(X, Y) = \sum_{i=1}^{n} x_i \, y_i$.

Обращение скалярного произведения в нуль служит признаком ортогональности (перпендикулярности) векторов.

Встретившись с вычислением отношений $f(x) / g(x)$ в ситуации, когда знаменатель обращается в нуль, не произносите глу-

пой фразы «*на ноль делить нельзя*», выясните, не обращается ли в нуль числитель, и при обнаружении такого явления вспомните о *правиле Лопиталя*

$$\lim_{x \to a} \frac{f(x)}{g(x)} = \lim_{x \to a} \frac{f'(x)}{g'(x)}.$$

Умножение матриц $C_{ml} = A_{mn} \cdot B_{nl}$ допустимо только в случае совпадения числа столбцов матрицы A и числа строк матрицы B:

$$c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk}; i = \overline{1, m}; k = \overline{1, l}.$$

Определитель (*детерминант*) *квадратной матрицы* $|A|$ – это число, сопоставляемое матрице A_{nn} по следующему правилу (разложение по строке):

$$\det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \cdot \det(A_{ij}^{*}) \quad ,$$

где A_{ij}^{*} – матрица, получаемая из A вычеркиванием строки i и столбца j . Детерминант скаляра равен его величине. При малых n ($n = 2$ или $n = 3$) удобнее использовать формулы

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc, \quad \begin{vmatrix} a & b & c \\ d & e & f \\ g & s & t \end{vmatrix} = aet + bfg + cds - ceg - bdt - afs \quad .$$

1.8. Вопросы для самоконтроля

1. Есть ли разница в представлении значений $0.5 \cdot 10^{-4}$, $0.005 \cdot 10^{-2}$, 0.00005, 0.000050?

2. Округлите до целого значения 3.524, 3.500, 4.500, 2.57.

3. Вычислите значение $F(a, b, c) = \frac{a+b}{(a-b)^2} \ln(a+c)$ при $a = 0.124$, $b = 0.121$, $c = 2.08$, запишите его, устранив неверные цифры, и выясните, с какой точностью следовало бы задать исходные параметры для гарантии пяти верных знаков результата.

4. Оценка по относительной погрешности, в отличие от абсолютной, разумна для величин, близких к единице или к 10^9 ?

5. На экран калькулятора выдано значение 12.333333. Если его относительная погрешности равна 0.001, то его запись с верными цифрами имеет вид 12.3300000, 12.33, 12.3 или 12.300000?

6. При вычислении синуса от малых значений аргумента абсолютная погрешность неизменна, удваивается или уменьшается вдвое?

7. С какой точностью достаточно задать число $\pi = 3.14159265358979323846\ldots$, чтобы найти объем шара с пятью верными знаками?

8. В одном из списков летописи, использованном одним из «птенцов гнезда Петрова» Василием Татищевым при написании «Истории Российской», сообщалось, что князь «…выступил в поход с **20000 воинов**», в другом списке – «**с 20 тысячами воинов**». Как следовало писать Татищеву, как вы воспринимаете эту информацию?

9. Для вычисления значения e^{-x} при x от 0 до 1 с погрешностью, не превышающей 0.01, с помощью разложения в ряд Тейлора необходимое число слагаемых равно 101, 5 или 21?

Глава 2. ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Рассмотрим систему n линейных алгебраических уравнений с n неизвестными

$$\begin{aligned} a_{11}\,x_1 + a_{12}\,x_2 \; + \ldots + a_{1n}\,x_n &= b_1, \\ a_{21}\,x_1 + a_{22}\,x_2 + \ldots + a_{2n}\,x_n &= b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_{n1}\,x_1 + a_{n2}\,x_2 + \ldots + a_{nn}\,x_n &= b_n. \end{aligned} \qquad (2.1)$$

Если обозначить через A матрицу коэффициентов при неизвестных, через B − вектор правой части ограничений и через X − вектор неизвестных, то систему (2.1) можно записать матричным уравнением

$$A\,X = B \qquad (2.2)$$

(далее под термином *вектор* мы будем подразумевать вектор-столбец и для действия транспонирования использовать символ **T** (в литературе и языках программирования часто используется знак апострофа).

Если эта система линейно независима (никакое из уравнений нельзя представить линейной комбинацией других), то ее **определитель отличен от нуля**, что обеспечивает **единственность решения**. В этом случае существует так называемая **обратная матрица** A^{-1} (аналогия с обычной скалярной величиной), для которой $A\;A^{-1} = A^{-1}\,A = E$, где E – единичная матрица (элементы главной диагонали равны 1 и остальные равны 0) и решение (2.2) можно представить в виде

$$X = A^{-1}\,B. \qquad (2.3)$$

Простейший (все познается в сравнении) метод решения системы (2.1) связан с **правилом Крамера***

$$x_j = \frac{\Delta_j}{\Delta},\; j = 1, 2, \ldots, n, \qquad (2.4)$$

где Δ − определитель (*детерминант*) матрицы A (главной матрицы); Δ_j − определитель матрицы (вспомогательной матрицы), получаемой из A заменой j-го столбца вектором B.

---

* Габриэль Крамер (1704−1752) − швейцарский математик, ученик и друг Иоганна Бернулли, один из создателей линейной алгебры.

Очевидно, что *при нулевом значении определителя главной матрицы система имеет множество решений или противоречива* (не имеет решения).

Пример. Рассмотрим систему трех уравнений

$$\begin{aligned} 3x_1 - x_2 &= 5,\\ -2x_1 + x_2 + x_3 &= 0,\\ 2x_1 - x_2 + 4x_3 &= 15. \end{aligned}$$

Находим определители:

$$\Delta = \begin{vmatrix} 3 & -1 & 0\\ -2 & 1 & 1\\ 2 & -1 & 4 \end{vmatrix} = 5,\ \Delta_1 = \begin{vmatrix} 5 & -1 & 0\\ 0 & 1 & 1\\ 15 & -1 & 4 \end{vmatrix} = 10,\ \Delta_2 = \begin{vmatrix} 3 & 5 & 0\\ -2 & 0 & 1\\ 2 & 15 & 4 \end{vmatrix} = 5,\ \Delta_3 = \begin{vmatrix} 3 & -1 & 5\\ -2 & 1 & 0\\ 2 & -1 & 15 \end{vmatrix} = 15,$$

откуда получаем $x_1 = 2$, $x_2 = 1$, $x_3 = 3$.

Другой путь решения систем связан с поиском обратной матрицы:

$$A^{-1} = \frac{\widehat{A}}{|A|},\quad \widehat{A} = \begin{vmatrix} A_{11} & A_{21} & \dots & A_{n1}\\ A_{12} & A_{22} & \dots & A_{n2}\\ \dots & \dots & \dots & \dots\\ A_{1n} & A_{2n} & \dots & A_{nn} \end{vmatrix},$$

где A_{ij} – так называемое алгебраическое дополнение соответствующего элемента матрицы (значение определителя матрицы, получаемой из A вычеркиванием строки i и столбца j, умноженное на $(-1)^{i+j}$).

Для вышеприведенного примера

$$A_{11} = (-1)^{1+1}\begin{vmatrix} 1 & 1\\ -1 & 4 \end{vmatrix} = 5,\quad A_{12} = (-1)^{1+2}\begin{vmatrix} -2 & 1\\ 2 & 4 \end{vmatrix} = 10,\quad A_{13} = (-1)^{1+3}\begin{vmatrix} -2 & 1\\ 2 & -1 \end{vmatrix} = 0,$$

$$A_{21} = (-1)^{2+1}\begin{vmatrix} -1 & 0\\ -1 & 4 \end{vmatrix} = 4,\quad A_{22} = (-1)^{2+2}\begin{vmatrix} 3 & 0\\ 2 & 4 \end{vmatrix} = 1,\quad A_{23} = (-1)^{2+3}\begin{vmatrix} 3 & -1\\ 2 & -1 \end{vmatrix} = 1,$$

$$A_{31} = (-1)^{3+1}\begin{vmatrix} -1 & 0\\ 1 & 1 \end{vmatrix} = -1,\quad A_{32} = (-1)^{3+2}\begin{vmatrix} 3 & 0\\ -2 & 1 \end{vmatrix} = -3,\quad A_{33} = (-1)^{3+3}\begin{vmatrix} 3 & -1\\ -2 & 1 \end{vmatrix} = 1$$

и

$$|A| = \begin{vmatrix} 3 & -1 & 0\\ -2 & 1 & 1\\ 2 & -1 & 4 \end{vmatrix} = 5,\quad A^{-1} = \begin{vmatrix} 1 & 0.8 & -0.2\\ 2 & 0.2 & -0.6\\ 0 & .2 & 0.2 \end{vmatrix},\ X = A^{-1}B = \begin{vmatrix} 2\\ 1\\ 3 \end{vmatrix}$$

Каким же из этих приемов пользоваться?

Правило Крамера требует вычисления $n + 1$ определителей порядка n (число арифметических операций превышает n^3 и при $n < 4$ не составляет затруднений), вычисление обратной матрицы – определителя порядка n, но зато n^2 определителей порядка $n - 1$ (здесь мы не останавливаемся на методах вычисления определителя, но напомним, что при $n > 3$ приходится использовать разложение по строке или столбцу, что сводит вычисление определителя порядка n к вычислению n определителей порядка $n - 1$). Близость определителя матрицы к нулю чревата значительной погрешностью.

Ниже мы рассмотрим другие, не столь дорогостоящие методы решения систем, поиска обратной матрицы и вычисления определителя.

2.1. Метод Гаусса

Школьник решает системы линейных уравнений путем подстановок (выражает некую переменную из какого-то уравнения, подставляет результат в остальные уравнения и т. д.), переписывая имена переменных счетное число раз и допуская счетное количество ошибок. Идея простая, но хаотичная и утомительная в реализации. Самым же популярным среди методов решения линейных систем является метод последовательного исключения переменных – метод Гаусса*, описанный во всех руководствах по численному анализу, реализуемый различными вычислительными схемами, менее утомительный и более изящный по сравнению с вышеприведенной реализацией.

Схема единственного деления для решения системы (2.1) предполагает выполнение следующего алгоритма.

---

* Иоганн Карл Фридрих Гаусс (1777–1855) − немецкий математик, астроном и физик, один из величайших математиков всех времен. С его именем связаны фундаментальные исследования почти во всех основных областях математики. Только за метод наименьших квадратов и нормальный закон распределения (фундамент современной статистики) Гаусса можно назвать «королем математиков».

Пусть $a_{11} \neq 0$. Первое уравнение делим на a_{11} и исключаем x_1 из последующих уравнений. Если коэффициент при x_2 во втором уравнении отличен от нуля, то второе уравнение делим на этот коэффициент и исключаем x_2 из последующих уравнений. Продолжая аналогичную процедуру, в результате получим систему с так называемой верхней треугольной матрицей коэффициентов при неизвестных

$$\begin{aligned} x_1 + c_{12}\,x_2 + c_{13}\,x_3 + \ldots + c_{1\,n-1}\,x_{n-1} + c_{1\,n}\,x_n &= d_1, \\ x_2 + c_{23}\,x_3 + \ldots + c_{2\,n-1}\,x_{n-1} + c_{2\,n}\,x_n &= d_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ x_{n\text{-}1} + c_{n-1\,n}\,x_n &= d_{n-1}, \\ x_n &= d_n. \end{aligned} \qquad (2.5)$$

Выполнив такую процедуру «прямого хода», затем выполняем «обратный ход» поиска значений переменных

$$x_n = d_n\,,\; x_k = d_k - \sum_{j=k+1}^{n} c_{kj}x_j\,,\; k = n-1, n-2, \ldots, 2, 1. \qquad (2.6)$$

В этом случае общее количество арифметических операций при $n > 7$ имеет порядок $N = \dfrac{2n(n+1)(n+2)}{3} + n(n-1) < n^3$, т. е. такой подход является менее трудоемким, чем правило Крамера. Решение ранее рассмотренного примера в этой схеме имеет вид:

3	−1	0	5
−2	1	1	0
2	−1	4	15

1	−1/3	0	5/3
0	1/3	1	10/3
0	−1/3	4	35/3

1	−1/3	0	5/3
0	1	3	10
0	0	5	15

1	−1/3	0	5/3
0	1	3	10
0	0	1	3

В результате получаем решение системы $x_3 = 3$,
$x_2 = 10 - (3)\,3 = 1$, $x_1 = 5/3 - (0)\,3 - (-1/3)\,1 = 2$.

Заметим, что **произведение** всех возникающих здесь **делителей** («ведущих элементов») **дает определитель** матрицы коэффициентов. Очевидно, что при обращении в нуль ведущих элементов приходится переставлять строки (уравнения) или столбцы (переменные).

Другая эквивалентная по затратам **схема полного исключения** отличается лишь тем, что исключение переменных выполняется не только из последующих, но и из предыдущих уравнений. Эта схема применима и для поиска обратной матрицы (если

к матрице $A \mid B$ присоединить единичную, то после действий полного исключения на месте матрицы $A \mid B \mid E$ получится матрица $E \mid X \mid A^{-1}$). Например:

$$\left(\begin{array}{ccc|c|ccc} 3 & -1 & 0 & 5 & 1 & 0 & 0 \\ -2 & 1 & 1 & 0 & 0 & 1 & 0 \\ 2 & -1 & 4 & 15 & 0 & 0 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c|ccc} 1 & -1/3 & 0 & 5/3 & 1/3 & 0 & 0 \\ 0 & 1/3 & 1 & 10/3 & 2/3 & 1 & 0 \\ 0 & -1/3 & 4 & 35/3 & -2/3 & 0 & 1 \end{array}\right) \Rightarrow$$

$$\Rightarrow \left(\begin{array}{ccc|c|ccc} 1 & 0 & 1 & 5 & 1 & 1 & 0 \\ 0 & 1 & 3 & 10 & 2 & 3 & 0 \\ 0 & 0 & 5 & 15 & 0 & 1 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c|ccc} 1 & 0 & 0 & 2 & 1 & 4/5 & -1/5 \\ 0 & 1 & 0 & 1 & 2 & 12/5 & -3/5 \\ 0 & 0 & 1 & 3 & 0 & 1/5 & 1/5 \end{array}\right)$$

В случае ведущих элементов, близких к нулю, погрешность вычислений значимо возрастает и часто, особенно в случае больших n, можно получить решение, далекое от истины. Во избежание такого эффекта практически во всех программных реализациях метода Гаусса используют **схему главных элементов**. Здесь сначала в качестве ведущего выбирается **наибольший по модулю** элемент матрицы коэффициентов, соответствующее уравнение делится на этот элемент и соответствующая переменная исключается из остальных уравнений. На следующих шагах решения за ведущий принимается наибольший по модулю элемент преобразованной матрицы без учета ранее рассмотренных уравнений (если ведущий элемент окажется нулевым, то либо система не имеет решений, либо система имеет бесчисленное множество решений). Например:

$$\left(\begin{array}{ccc|c} 3 & -1 & 0 & 5 \\ -2 & 1 & 1 & 0 \\ 2 & -1 & \mathbf{4} & 15 \end{array}\right) \quad \left(\begin{array}{ccc|c} \mathbf{3} & -1 & 0 & 5 \\ -5/2 & 5/4 & 0 & -15/4 \\ 1/2 & -1/4 & 1 & 15/4 \end{array}\right) \quad \left(\begin{array}{ccc|c} 1 & -1/3 & 0 & 5/3 \\ 0 & \mathbf{5/12} & 0 & 5/12 \\ 0 & -1/12 & 1 & 35/12 \end{array}\right) \quad \left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 3 \end{array}\right)$$

2.2. Разложение матрицы в произведение треугольных (LU-факторизация) и метод Краута − Дулитла

Идея LU-факторизации сводится к представлению матрицы A произведением нижней и верхней треугольных матриц $A = L \cdot U$, где

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}. \qquad (2.7)$$

Выполняя их умножение, приходим к системе n^2 уравнений

$$a_{ij} = \sum_{k=1}^{l} l_{ik}\, u_{kj};\ i, j = 1, 2, \dots, n$$

с $n^2 + n$ неизвестными, которую можно решить однозначно лишь задав n неизвестных равными каким-то константам (например, равными 1). Такое разложение в литературе называют схемой Халецкого (Холецкого?), а предложенные в 1941 г. на ее основе методы решения системы $A\,X = B$ методом Краута (элементы главной диагонали матрицы U равны единице) или методом Дулитла (главная диагональ матрицы L единичная) [8].

Возьмем для примера систему $A\,X = D$ и разложение $A = B \cdot C$, где

$$B = \begin{bmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & c_{12} & \dots & c_{1n} \\ 0 & 1 & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \qquad (2.8)$$

Перемножая эти матрицы, мы получаем систему n^2 уравнений с n^2 неизвестными, решение которой можно отыскать в виде

$$b_{i1} = a_{i1}\ (i = 1, 2, \dots, n);\ c_{1j} = a_{1j} / b_{11}\ (j = 2, 3, \dots, n);$$

$$c_{ij} = \frac{1}{b_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} b_{ik}\, c_{kj}\right) \quad (i = 2, 3, \dots, j-1);$$

$$b_{ij} = a_{ij} - \sum_{k=1}^{i-1} b_{ik}\, c_{kj}\ (i \geq j) \qquad (2.9)$$

(находим первый столбец матрицы B и первую строку C, затем второй столбец B и вторую строку C и т. д.).

Так при $n = 3$ последовательно находим значения

$$b_{11} = a_{11}, \quad b_{21} = a_{21}, \quad b_{31} = a_{31},$$
$$c_{11} = 1, \quad c_{12} = a_{12}/b_{11}, \quad c_{13} = a_{13}/b_{11},$$
$$b_{22} = a_{22} - b_{21}\, c_{12}\ , \ b_{32} = a_{32} - b_{31}\, c_{12},$$
$$c_{22} = 1, \quad c_{23} = \tfrac{1}{b_{22}}(c_{23} - b_{21}\, c_{13}),$$
$$b_{33} = a_{33} - b_{31}\, c_{13} - b_{32}\, c_{23}, \quad \mathrm{c}_{33} = 1.$$

Для матрицы из рассмотренного выше примера имеем

$$A = \left|\begin{array}{rrr} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{array}\right| \begin{array}{l} b_{11} = 3, \quad b_{21} = -2, \quad b_{31} = 2, \\ c_{11} = 1, \quad c_{12} = -1/3, \quad c_{13} = 0, \\ b_{22} = 1-(-2)\cdot(-1/3) = 1/3, \ b_{32} = -1-(2)\cdot(-1/3) = -1/3, \\ c_{22} = 1, \ c_{23} = [1-(-2)\cdot(0)] / (1/3) = 3, \\ b_{33} = 4 - (2)\cdot(0) - (-1/3)\cdot(3) = 5, \ c_{33} = 1. \end{array}$$

Отсюда

$$\left|\begin{array}{rrr} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{array}\right| = \left|\begin{array}{rrr} 3 & 0 & 0 \\ -2 & 1/3 & 0 \\ 2 & -1/3 & 5 \end{array}\right| \cdot \left|\begin{array}{rrr} 1 & -1/3 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{array}\right|$$

Получив такое разложение, имеем $B \cdot C \cdot X = D$ и, выполнив замену $C \cdot X = Y$, приходим к системе двух уравнений с треугольными матрицами коэффициентов $B \cdot Y = D$ и $C \cdot X = Y$, решение которых достаточно просто

$$y_1 = \frac{d_1}{b_{11}}, \ y_i = \frac{1}{b_i}\left(d_i - \sum_{k=1}^{i-1} b_{ik}\, y_k\right), \ i = 2, 3, \ldots, n;$$
$$x_n = y_n, \ x_i = y_i - \sum_{k=i+1}^{n} c_{ik}\, x_k, \quad i = n-1, n-2, \ldots, 1. \qquad (2.10)$$

Так для нашего примера возникают системы

$$\left|\begin{array}{rrr} 3 & 0 & 0 \\ -2 & 1/3 & 0 \\ 2 & -1/3 & 5 \end{array}\right| \left|\begin{array}{c} y_1 \\ y_2 \\ y_3 \end{array}\right| = \left|\begin{array}{c} 5 \\ 0 \\ 15 \end{array}\right|, \quad \left|\begin{array}{rrr} 1 & -1/3 & 0 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{array}\right| \left|\begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array}\right| = \left|\begin{array}{c} y_1 \\ y_2 \\ y_3 \end{array}\right|$$

откуда получаем

$$y_1 = 5/3, \ y_2 = [0 - (-2)\cdot(5/3)] / (1/3) = 10,$$
$$y_3 = [15 - (2)\cdot(5/3) - (-1/3)\cdot(10)] / 5 = 3,$$
$$x_3 = 3, \ x_2 = 10 - (3)\cdot(3) = 1, \ x_1 = 5/3 - (-1/3)\cdot(1) - (0)\cdot(3) = 2.$$

Количество арифметических операций при реализации метода Краута имеет тот же порядок, что и метода Гаусса.

В среде MatLab можно воспользоваться стандартной процедурой `[L,U,P]=lu(A)`, где за единичную принята главная диагональ первого сомножителя (вывод матриц может сопровождаться перестановкой строк / столбцов).

2.3. Метод квадратных корней

В многочисленных приложениях (в частности, при обработке статистических данных) возникает система уравнений с симметрической матрицей коэффициентов (матрица A называется симметрической, если для ее элементов выполняется условие $a_{ij} = a_{ji}$ при всех i и j). Такие матрицы представляют частный случай так называемых эрмитовых матриц*, в которых элементы главной диагонали вещественны, а симметричные образуют пары комплексно сопряженных чисел. Например:

$$\begin{bmatrix} 1 & 8 & 2+i & 0 \\ 8 & 3 & 11 & i \\ 2-i & 11 & 7 & 1 \\ 0 & -i & 1 & 9 \end{bmatrix}.$$

Всякую вещественную симметрическую матрицу можно представить произведением взаимно транспонированных матриц

$$A = R^{\mathrm{T}} R, \tag{2.11}$$

где

$$R = \begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & r_{22} & \ldots & r_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & r_{nn} \end{bmatrix}. \tag{2.12}$$

Перемножая (2.11) матрицы, получаем систему

$$\begin{aligned} &r_{1i}^2 + r_{2i}^2 + \ldots + r_{ii}^2 = a_{ii};\ i,\ j = 1, 2, \ldots, n; \\ &r_{1i} r_{1j} + r_{2i} r_{2j} + \ldots + r_{ii} r_{ij} = a_{ij},\ i < j, \end{aligned} \tag{2.13}$$

---

* Шарль Эрмит (1822–1901) – выдающийся французский математик второй половины XIX века. Наиболее известны сегодня его работы в теории квадратичных форм, ортогональных многочленов (многочлены Эрмита), эллиптических функций и алгебре.

решение которой имеет вид

$$r_{ii} = \sqrt{a_{i\,i} - \sum_{k=1}^{i-1} r_{ki}^2}\ ,\ i = 1, 2, \ldots, n;$$
$$r_{ij} = \frac{1}{r_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} r_{ki}\, r_{kj} \right),\ j = i+1, i+2, \ldots, n. \qquad (2.14)$$

Полученное разложение сводит систему $A\,X = B$ к двум системам с треугольными матрицами коэффициентов $R^{\mathrm{T}}\,Y = B$, $R\,X = Y$, решаемым по схеме, описанной выше для метода Краута.

Если матрица A − положительно определенная (все ее главные миноры положительны), все элементы матрицы R − вещественные числа. В общем случае элементы R могут быть и чисто мнимыми числами. Самый плохой случай в технологии такого разложения возникает, когда какие-то из главных миноров обращаются в нуль.

Метод квадратных корней является самым быстродействующим, но его программная реализация требует операций с повышенной точностью (точность теряется из-за операций извлечения корня) и арифметики комплексных чисел.

В среде MatLab обращение к функции `[R,p]=chol(A)` может завершиться прерыванием с сообщением `??? Error using ==> chol \ Matrix must be positive definite` (матрица должна быть положительно определенной).

Рассмотрим пример применения метода квадратных корней.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 7 \end{bmatrix},\quad B = \begin{bmatrix} 5 \\ 8 \\ 13 \end{bmatrix}.$$

Получаем

$$r_{11} = \sqrt{a_{11}} = \sqrt{1} = 1,\ r_{12} = \frac{a_{12}}{r_{11}} = \frac{2}{1} = 2\,,\ r_{13} = \frac{a_{13}}{r_{11}} = \frac{3}{1} = 3,$$
$$r_{22} = \sqrt{a_{22} - r_{12}^2} = \sqrt{3 - 2^2} = \sqrt{-1} = i,$$
$$r_{23} = \frac{1}{i}\left(a_{23} - r_{12} r_{13}\right) = \frac{1}{i}\left(4 - 2 \cdot 3\right) = -2 / i = 2\,i,$$
$$r_{33} = \sqrt{a_{33} - r_{13}^2 - r_{23}^2} = \sqrt{7 - 3^2 - (2\,i)^2} = \sqrt{-2 + 4} = \sqrt{2}.$$

То есть

$$R = \begin{vmatrix} 1 & 2 & 3 \\ 0 & i & 2\,i \\ 0 & 0 & \sqrt{2} \end{vmatrix}.$$

Согласно методу Краута, получаем

$$R^{\mathrm{T}} Y = B; \quad \begin{vmatrix} 1 & 0 & 0 \\ 2 & i & 0 \\ 3 & 2\,i & \sqrt{2} \end{vmatrix} \cdot \begin{vmatrix} y_1 \\ y_2 \\ y_3 \end{vmatrix} = \begin{vmatrix} 5 \\ 8 \\ 13 \end{vmatrix}$$

$y_1 = 5 / 1 = 5$

$y_2 = (8 - 2 \cdot 5) / i = -2 / i = 2\,i$

$y_3 = (13 - 3 \cdot 5 - (2\,i) \cdot (2\,i)) / (\sqrt{2}) = 2 / \sqrt{2} = \sqrt{2}$

$$C \cdot X = Y; \quad \begin{vmatrix} 1 & 2 & 3 \\ 0 & i & 2\,i \\ 0 & 0 & \sqrt{2} \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = \begin{vmatrix} 5 \\ 2\,i \\ \sqrt{2} \end{vmatrix}$$

$x_3 = \sqrt{2} / \sqrt{2} = 1$

$x_2 = 2\,i - 2\,i \cdot 1 = 0$

$x_1 = 5 - 3 \cdot 1 - 2 \cdot 0 = 2$

2.4. Метод прогонки для систем с трехдиагональной матрицей коэффициентов*

В многочисленных приложениях (конечноразностная аппроксимация, краевые задачи для дифференциальных уравнений, уравнения математической физики и др.) приходится иметь дело с системами уравнений следующей структуры:

$$\begin{vmatrix} b_1 & c_1 & 0 & 0 & 0 & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & 0 & \dots & 0 & 0 \\ 0 & a_3 & b_3 & c_3 & 0 & \dots & 0 & 0 \\ 0 & 0 & a_4 & b_4 & c_4 & .... & 0 & 0 \\ 0 & 0 & 0 & a_k & b_k & ... & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & & a_n & b_n \end{vmatrix} \cdot \begin{vmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \\ x_n \end{vmatrix} = \begin{vmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ \\ d_n \end{vmatrix}$$

или в компактном виде

$$\begin{gathered} b_1 x_1 + c_1 x_2 = d_1, \\ a_k x_{k-1} + b_k x_k + c_k x_{k+1} = d_k,\ k = 2, ..., n - 1, \\ a_n x_{n-1} + b_n x_n = d_n. \end{gathered} \qquad (2.15)$$

Положим

$$x_k = P_k x_{k+1} + Q_k,\ k = 1, 2, \dots, n - 1 \qquad (2.16)$$

и, подставив в (2.15), получим

---

* Метод прогонки предложен одним из крупнейших математиков XX века И. М. Гельфандом (1913–2009) и О. В. Локуциевским в 1952 г., опубликован в 1960 г. и стал популярным благодаря публикациям [2, 9].

$$a_k\,(P_{k-1}\,x_k + Q_{k-1}) + b_k x_k + c_k x_{k+1} = d_k,\ k = 2, 3, \ldots, n-1$$

или

$$(a_k P_{k-1} + b_k)\,x_k + c_k\,x_{k+1} = d_k - a_k\,Q_{k-1},\ k = 2, 3, \ldots, n-1.$$

С учетом представления первого уравнения в виде

$$x_2 = (d_1 - b_1\,x_1) / c_1$$

и последнего:

$$a_n\,(P_{n-1}\,x_n + Q_{n-1}) + b_n\,x_n = d_n,$$

получаем двухшаговую процедуру решения: на *прямом* ходе ищем прогоночные коэффициенты (2.16) в порядке роста k и на *обратном* – находим решение системы:

$$P_1 = -c_1 / b_1,\ Q_1 = d_1 / b_1,$$

$$P_k = -\frac{c_k}{b_k + a_k P_{k-1}},\ Q_k = \frac{d_k - a_k Q_{k-1}}{b_k + a_k P_{k-1}},\ k = 2, \ldots, n-1; \qquad (2.17)$$

$$x_n = \frac{d_n - a_n Q_{n-1}}{b_n + a_n P_{n-1}},\ x_k = P_k x_{k+1} + Q_k,\ k = n-1, n-2, \ldots, 1.$$

Заметим, что трехдиагональная матрица коэффициентов содержит только $3n - 2 << n^2$ ненулевых элементов, что при размерности порядка сотен существенно сокращает требования к объему используемой памяти компьютера. Этот метод называют *методом прогонки*; его использование значимо уменьшает объем вычислительных затрат в сравнении с любыми другими методами.

Существуют видоизменения метода прогонки для систем с более чем трехдиагональной матрицей коэффициентов (в частности, пятидиагональных), но здесь приходится обращать матрицы.

2.5. Итерационные методы

Как мы отмечали выше, при использовании *прямых методов* решения задач предсказуемо число необходимых вычислительных действий (поиск значения полинома, вычисление определителя через разложение по строке или столбцу, решение системы линейных алгебраических уравнений методом Гаусса или методом квадратных корней). Увы, при решении задач большой раз-

мерности никакие ухищрения не спасут от быстрого роста погрешности и хорошо, если полученные оценки правдоподобны (неверующему читателю предлагаем найти обратную к матрице 200-го порядка). Это и вынуждает использовать методы последовательных приближений (*методы итераций*).

Их идея сводится к выбору начального приближения к искомому решению и последующему последовательному его уточнению до выполнения какого-то критерия или установления факта нереальности выполнения этого критерия (таким подходом мы пользовались выше при вычислении суммы членов ряда Тейлора и поиске $y = \sqrt[k]{x}$). Несомненным достоинством итерационных процедур является возможность получения результата с любой требуемой точностью и их устойчивость к промежуточным ошибкам.

2.5.1. Метод простой итерации

Идея итерационных методов решения системы уравнений $A\,X = B$ состоит в преобразовании ее к виду

$$X = \alpha\, X + \beta \qquad (2.18)$$

с последующим использованием сходящегося итерационного процесса

$$X^{(k+1)} = \alpha\, X^{(k)} + \beta,\ k = 0, 1, 2, \ldots, \qquad (2.19)$$

где начальное приближение $X^{(0)}$ выбирается произвольно, например, равным нулю или оценкам, найденным другими методами. Процесс итераций заканчивается обнаружением близости очередных приближений.

Прежде чем говорить об условиях сходимости итерационного процесса (2.19), напомним понятие *нормы матрицы*. *Нормой* элемента U называют скалярную величину $\|U\|$, обладающую свойствами:

1) $\|U\| \geq 0$ при всех U, причем $\|U\| = 0$ при $U = 0$;
2) $\|k \cdot U\| = k \cdot \|U\|$, где k – скаляр; (2.20)
3) $\|U + V\| \leq \|U\| + \|V\|$.

С этим понятием мы встречаемся в курсе школьной математики при знакомстве с длиной вектора, заявляя, что в треугольнике

длина любой стороны меньше суммы длин двух других его сторон.

За норму скаляра принимается его абсолютная величина. В качестве нормы вектора приемлемы варианты

$$\|X\|_1 = \sum_i |x_i|,\ \|X\|_2 = \sum_i x_i^2,\ \|X\|_{\inf} = \max_i |x_i|. \qquad (2.21)$$

За норму матрицы выбирают из нескольких вариантов:

$$\|A\|_1 = \max_j \sum_i |a_{ij}|,\ \|A\|_{\inf} = \max_i \sum_j |a_{ij}|,\ \|A\|_2 = \sqrt{\lambda_{\max}(A\,A^{\mathrm{T}})} \qquad (2.22)$$

(здесь $\lambda_{\max}$ – максимальное собственное число матрицы $A \cdot A^{\mathrm{T}}$; о поиске собственных чисел см. ниже) или завышенную евклидову норму

$$\|A\| = \sqrt{\sum_i \sum_j a_{ij}^2}\,. \qquad (2.23)$$

Достаточным условием сходимости итерационного процесса (2.19) является весьма жесткое требование:

$$\|\alpha\| \le K < 1. \qquad (2.24)$$

Выполнение этого условия легко обеспечивается, если внедиагональные элементы матрицы A по модулю много меньше соответствующих диагональных элементов $|a_{ij}| << |a_{ii}|$, например, при всех i

$$\sum_{\substack{j=1 \\ i \ne i}}^{n} |a_{ij}| < a_{ii}\,. \qquad (2.25)$$

В общем случае сведение $A\,X = B$ к виду $X = \alpha\,X + \beta$ с соблюдением условия $\|\alpha\| \le K < 1$ неоднозначно и требует определенного искусства.

Пример 1. Возьмем такую систему [4]:

$$4\,x_1 + 0.24\,x_2 - 0.08\,x_3 = 8,$$
$$0.09\,x_1 + 3\,x_2 - 0.15\,x_3 = 9,$$
$$0.04\,x_1 - 0.08\,x_2 + 4\,x_3 = 20$$

и элементарным делением на диагональные элементы преобразуем ее к эквивалентной системе

$$x_1 = 2 - 0.06\,x_2 + 0.02\,x_3,$$
$$x_2 = 3 - 0.03\,x_1 + 0.05\,x_3,$$
$$x_3 = 5 - 0.01\,x_1 + 0.02\,x_2.$$

При нулевом начальном приближении получаем

$$X^{(1)} = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}; \quad X^{(2)} = \begin{bmatrix} 0 & -0.06 & 0.02 \\ -0.03 & 0 & 0.05 \\ -0.01 & 0.02 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 1.9200 \\ 3.1900 \\ 5.0400 \end{bmatrix}$$

$$X^{(3)} = \begin{bmatrix} 0 & -0.06 & 0.02 \\ -0.03 & 0 & 0.05 \\ -0.01 & 0.02 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1.9200 \\ 3.1900 \\ 5.0400 \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} \mathbf{1.9}094 \\ \mathbf{3.19}44 \\ \mathbf{5.04}46 \end{bmatrix}; X^{(4)} = \begin{bmatrix} \mathbf{1.909}23 \\ \mathbf{3.194}95 \\ \mathbf{5.044}85 \end{bmatrix}$$

(здесь сходимость процесса очевидна; уже найденное четвертое приближение гарантирует, по крайней мере, четыре верных значащих цифры).

Заметим, что соблюдения условия $\|\alpha\| \le K < 1$ достаточно для сходимости, но не всегда в этом есть необходимость.

Пусть исходная система приведена к виду

$$x_1 = 1.5 + 0.5\,x_1 - x_2,$$
$$x_2 = 1.1 + 0.5\,x_1 - 0.6\,x_2.$$

Приведенные в (2.22) оценки нормы матрицы коэффициентов $\|A\|_1 = 1.6$, $\|A\|_{\inf} = 1.1$, $\|A\|_2 = 1.84$

(последнюю из них обсудим позднее) превышают 1. Тем не менее процесс итераций оказывается сходящимся (см. таблицу ниже).

Номер итерации	1	2	3	4	5	6	7	8	9
x_1	1.50	1.15	0.89	0.98	1.023	1.001	0.995	1.000	1.001
x_2	1.10	1.19	0.96	0.97	1.011	1.006	0.997	0.999	1.001

Условие $|a_{ij}| << |a_{ii}|$ эффективного преобразования $A\,X = B$ к виду $X = \alpha\,X + \beta$ делением на диагональные элементы можно ослабить, если диагональные элементы близки к 1, а внедиагональные достаточно малы.

Пример 2. Возьмем систему

$$1.12\,x_1 + 0.24\,x_2 - 0.08\,x_3 = 8,$$
$$0.09\,x_1 + 0.95\,x_2 - 0.15\,x_3 = 9,$$
$$0.04\,x_1 - 0.08\,x_2 + 1.53\,x_3 = 20$$

и преобразуем ее к виду

$$x_1 = 8 - 0.12\,x_1 - 0.24\,x_2 + 0.08\,x_3,$$
$$x_2 = 9 - 0.09\,x_1 + 0.05\,x_2 + 0.15\,x_3,$$
$$x_3 = 20 - 0.04\,x_1 - 0.08\,x_2 - 0.53\,x_3.$$

Очевидно, что норма матрицы

$$\alpha = \begin{vmatrix} -0.12 & -0.24 & 0.08 \\ -0.09 & 0.05 & 0.15 \\ -0.04 & -0.08 & -0.53 \end{vmatrix}$$

меньше 1 и процесс простой итерации сходится

k	1	2	3	4	5	6	7	8	9
x_1	8	6.48	5.08	6.08	5.49	5.80	5.64	5.72	5.68
x_2	9	11.73	10.26	11.21	10.72	10.97	10.84	10.91	10.88
x_3	20	8.36	14.37	11.36	12.84	12.12	12.47	12.30	12.38

Процесс итераций заканчивается обнаружением близости очередных приближений в смысле абсолютной или относительной погрешности [12, с. 78]:

$$\max_i \left|x_i^{(k+1)} - x_i^k\right| < \varepsilon, \quad \frac{\|a\|_1}{1-\|a\|_1} \max_i \left|x_i^{(k+1)} - x_i^k\right| < \varepsilon,$$
$$\max_i \left|x_i^{(k+1)} - x_i^k\right| < \varepsilon \left|x_i^{(k+1)}\right|.$$

2.5.2. Метод Зейделя*

Процесс простой итерации (2.19) в развернутом виде

$$x_i^{(k+1)} = \sum_{j=1}^{n} \alpha_{i\,j}\, x_j^{(k)} + \beta_i,\ i = 1, 2, ..., n.$$

Метод Зейделя отличается тем, что на очередной итерации берут не оценки предыдущей итерации, а последние полученные:

---

* Филипп Людвиг фон Зейдель (1821–1896) – немецкий математик и астроном, автор приведенного здесь итерационного метода.

$$x_1^{(k+1)} = \sum_{j=1}^{n} \alpha_{i\,j}\, x_j^{(k)} + \beta_1,$$

$$x_2^{(k+1)} = \alpha_{21} x_1^{(k)} + \sum_{j=2}^{n} \alpha_{2\,j} x_j^{(k)} + \beta_2, \qquad (2.26)$$

$$x_i^{(k+1)} = \sum_{j=1}^{i-1} \alpha_{ij} x_j^{(k+1)} + \sum_{j=i}^{n} \alpha_{ij} x_j^{(k)} + \beta_i.$$

Так для приведенной выше системы

$$x_1 = 2 - 0.06\, x_2 + 0.02\, x_3,$$
$$x_2 = 3 - 0.03\, x_1 + 0.05\, x_3,$$
$$x_3 = 5 - 0.01\, x_1 + 0.02\, x_2.$$

при нулевом начальном приближении получаем

$$x_1^{(1)} = 2,$$
$$x_2^{(1)} = 3 - 0.03 \cdot 2 = 2.94,$$
$$x_3^{(1)} = 5 - 0.01 \cdot 2 + 0.02 \cdot 2.94 = 5.0388,$$
$$x_1^{(2)} = 2 - 0.06 \cdot 2.94 + 0.02 \cdot 5.0388 = 1.9244, \ldots$$

Как правило, метод Зейделя ускоряет сходимость приближений к истинному решению.

2.5.3. Метод релаксации

Среди итерационных методов достаточно популярен *метод релаксации* (*ослабления*) [2], где на каждом шаге меняется значение только одной из компонент вектора приближения. Здесь исходную систему $A\,X = B$ (например, делением на диагональные элементы) преобразуем к эквивалентному виду

$$\begin{aligned} &-x_1 + c_{12}\, x_2 + \ldots + c_{1n}\, x_n + d_1 = 0, \\ &c_{21}\, x_1 - x_2 + \ldots + c_{2n}\, x_n + d_2 = 0, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &c_{n1}\, x_1 + c_{n2}\, x_2 + \ldots - x_n + d_n = 0 \end{aligned} \qquad (2.27)$$

и подстановкой начального приближения $X^{(0)}$ в (2.27) получаем так называемые «невязки»:

$$\begin{aligned} R_1^{(0)} &= d_1 - x_1^{(0)} + c_{12}\, x_2^{(0)} + \ldots + c_{1n}\, x_n^{(0)}, \\ R_2^{(0)} &= d_2 - x_2^{(0)} + c_{21}\, x_1^{(0)} + \ldots + c_{2n}\, x_n^{(0)}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ R_n^{(0)} &= d_n - x_n^{(0)} + c_{n1}\, x_1^{(0)} + \ldots + c_{n\,n-1}\, x_{n-1}^{(0)} \end{aligned} \qquad (2.28)$$

или в компактной форме

$$R_i^{(0)} = d_i - x_i^{(0)} + \sum_{j \neq i} c_{ij}\, x_j^{(0)}, \quad i = 1, 2, \ldots, n. \qquad (2.29)$$

Выберем максимальную по модулю невязку $R_s^{(0)}$ и положим в очередном приближении

$$x_s^{(1)} = x_s^{(0)} + R_s^{(0)} \qquad (2.30)$$

(подавляя эту невязку). Корректируя остальные невязки, имеем

$$R_s^{(1)} = 0,\ R_i^{(1)} = R_i^{(0)} + c_{is} \cdot R_s^{(0)},\ i \neq s. \qquad (2.31)$$

Среди найденных невязок вновь отыскиваем наибольшую по модулю $R_k^{(1)}$ и, положив $x_k^{(2)} = x_k^{(1)} + R_k^{(1)}$, получаем очередные $R_k^{(2)} = 0$, $R_i^{(2)} = R_i^{(1)} + c_{is} \cdot R_k^{(1)}$, $i \neq k$, и т. д. до получения максимума невязки в пределах заданной точности.

Пример. Обратимся к рассмотренной выше системе:

$$4\,x_1 + 0.24\,x_2 - 0.08\,x_3 = 8,$$
$$0.09\,x_1 + 3\,x_2 - 0.15\,x_3 = 9,$$
$$0.04\,x_1 - 0.08\,x_2 + 4\,x_3 = 20.$$

Разделив на элементы главной диагонали, имеем

$$x_1 + 0.06\,x_2 - 0.02\,x_3 = 2,$$
$$0.03\,x_1 + 3\,x_2 - 0.05\,x_3 = 3,$$
$$0.01\,x_1 - 0.02\,x_2 + x_3 = 5.$$

Преобразуем к виду, приемлемому для последующих итераций:

$$R_1 = 2 - x_1 - 0.06\,x_2 + 0.02\,x_3,$$
$$R_2 = 3 - 0.03\,x_1 - x_2 + 0.05\,x_3,$$
$$R_3 = 5 - 0.01\,x_1 + 0.02\,x_2 - x_3.$$

Задав $x_1^{(1)} = 2$, $x_2^{(1)} = 3$, $x_3 = 5$, получаем

$$R_1^{(2)} = -0.08,\ R_2^{(2)} = 0.19,\ R_3^{(2)} = 0.04.$$

Выбираем наибольшую невязку $R_2^{(2)} = 0.19$ и, согласно (2.30), корректируем: $\boldsymbol{x_2^{(2)} = 3 + 0.19 = 3.19}$.

Очевидно $R_2^{(2)} = 0$,

$$R_1^{(2)} = R_1^{(1)} - 0.06\,R_2^{(1)} = -0.08 - 0.06 \cdot 0.19 = -0.0912,$$
$$R_3^{(2)} = R_3^{(1)} + 0.02\,R_2^{(1)} = 0.04 + 0.02 \cdot 0.19 = 0.0438.$$

Наибольшая невязка $R_1^{(2)} = -0.0912$ и $\boldsymbol{x_1^{(3)} = 2 - 0.0912 = 1.9088}$.

Отсюда $R_1^{(3)} = 0$,

$$R_2^{(3)} = R_2^{(2)} - 0.03\,R_1^{(2)} = 0 - 0.03\,(-0.0912) = 0.002736,$$
$$R_3^{(3)} = R_3^{(2)} - 0.01\,R_1^{(2)} = 0.0438 - 0.01\,(-0.0912) = 0.044712.$$

Наибольшая невязка $R_3^{(3)} = 0.044712$, $\boldsymbol{x_3^{(3)} = 5 + 0.044712 = 5.044712}$ и т. д.

Нетрудно увидеть, что объем вычислительной работы не больше, чем при использовании других итерационных методов.

2.6. Коротко о других линейных системах и методах

Можно свести решение системы линейных уравнений

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \; i = \overline{1, n} \tag{2.32}$$

к задаче минимизации (до нуля) функции

$$\sum_{i=1}^{n} g_i \left(\sum_{j=1}^{n} a_{ij} x_j - b_i \right)^2, \; g_i > 0, i = \overline{1, \; n}. \tag{2.33}$$

В самом деле, если решение (2.32) существует и оно единственно, то (2.33) обращается в нуль. В случае противоречивой системы достижение нуля нереально, и существенную роль играет выбор весовых коэффициентов g_i, выражающих степень значимости соблюдения того или иного уравнения (2.32). Очевидно, что решение задачи оптимизации не проще использования других рассмотренных здесь методов, требует знакомства с соответствующими методами оптимизации и, может быть, даже с методами случайных испытаний (методами Монте-Карло).

Мы не пытаемся рассмотреть все многообразие прямых и итерационных методов решения линейных алгебраических систем, большинство из которых потеряло свою практическую значимость в век могущества вычислительной техники. Заметим только, что для систем высокого порядка, где матрица коэффициентов может быть разбита на блоки, среди которых есть нулевые, используют *клеточные методы*, сводящие исходную задачу к последовательному перебору клеток (блоков) матрицы и поиску соответствующих обратных матриц.

Особого внимания заслуживают и «*плохо обусловленные*» системы, для которых определитель матрицы близок к нулю. Здесь малейшее изменение исходных данных ведет к значительному изменению решения (решение неустойчиво по исходным

данным). В принципе может обнаружиться и отсутствие решения − несовместность системы. Разрешение такой ситуации, связанной обычно с некорректной постановкой задачи, предлагает *метод регуляризации* А. Н. Тихонова*, сводящий исходную систему $A\,X = B$ к всегда совместной системе

$$(A^{\mathrm{T}}A + \alpha\, E)\, X = A^{\mathrm{T}}B,\ \alpha > 0, \qquad (2.34)$$

где параметр α выбирается с учетом требуемой точности решения и погрешности исходных данных.

К решению системы $A^{\mathrm{T}}A\,X = A^{\mathrm{T}}B$ сводится и *случай переопределенных систем,* в которых число уравнений m превышает число переменных n ($m > n$).

2.7. Решение систем линейных уравнений в среде MatLab

Отличительной способностью системы MatLab является возможность выполнения операций не только над скалярными величинами, но и над массивами (векторами, матрицами). Так для решения системы $A\,X = B$ достаточно задать массив-матрицу A и массив-столбец B командами

```
» A = [3 −1 0; −2 1 1; 2 −1 4];
» B = [5; 0; 15];
```

и выполнить команду `X=A\B` или `X=B/A` (иллюзия привычной для дилетанта операции деления), получая столбец решений X.

Запись `A'` определяет транспонирование.

В библиотеке MatLab для решения других задач линейной алгебры можно выделить следующие функции:

1) `det(A)` – возвращает определитель (для квадратной матрицы);

---

* Андрей Николаевич Тихонов (1906–1993) – выдающийся советский математик и геофизик. Область научных интересов – топология и функциональный анализ, дифференциальные и интегральные уравнения, математическая физика и вычислительная математика, обратные и некорректно поставленные задачи, возникающие при изучении физики плазмы, в геофизике, электродинамике и др. В эпоху создания ЭВМ и начала бурного развития численного анализа сделал многое для подготовки квалифицированных специалистов по вычислительной и прикладной математике и кибернетике.

2) `rank(A)` – вычисляет ранг матрицы (число линейно независимых строк / столбцов);

3) `norm(A)` – возвращает норму матрицы (2.22) $\|A\|_2$;

4) `inv(A)`, `A^(-1)` – предназначена для обращения матрицы (при близости матрицы к вырожденной выдаются предупреждения о ненадежности результатов); для решения системы $A\,X = B$ можно использовать оператор `X = inv(A)*B`;

5) `R=chol(A)` – позволяет получить разложение $R^{\mathrm{T}}R$ положительно определенной симметрической действительной или комплексной эрмитовой (если $A_{ij} = a + b_i$, то $A_{ji} = a - b_i$) матрицы A; R – верхняя треугольная матрица. Например, задав массив A и выполнив `R=chol(A)`, получаем

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

6) `[L,U]=lu(A)` – задает разложение произвольной квадратной матрицы $A = L\,U$ в произведение нижней и верхней треугольных матриц

$$L = \begin{bmatrix} 1.000 & 0 & 0 & 0 \\ 1.000 & 0.333 & 1.000 & 1.000 \\ 1.000 & 0.667 & 1.000 & 0 \\ 1.000 & 1.000 & 0 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 1.000 & 1.000 & 1.000 & 1.000 \\ 0 & 3.000 & 9.000 & 19.000 \\ 0 & 0 & -1.000 & -3.667 \\ 0 & 0 & 0 & 0.333 \end{bmatrix}.$$

Здесь матрицы L и U выводятся с точностью до перестановки строк;

7) `X=nnis(A,B)` позволяет искать решение *переопределенной системы* $A\,X = B$ методом наименьших квадратов, когда отыскиваются *неотрицательные решения* X, минимизирующие выражение norm($A\,X - B$).

2.8. Вопросы для самоконтроля

1. Каково условие единственности решения системы $A\,X = B$? Сколько решений может иметь такая система?

2. В чем суть гауссовой схемы главных элементов? Дайте обоснование ее целесообразности.

3. Когда имеет смысл использовать метод квадратных корней? Представьте себе, что матрица A при вводе задана верхним треугольником в виде одномерного массива AA. Как ссылку $A[i, j]$ заменить при выборке из AA?

4. Почему при решении задачи методом квадратных корней требуют задания данных с двойной точностью?

5. В чем отличие методов простой итерации и Зейделя?

6. Какова принципиальная разница между методом Гаусса и итерационными методами?

7. Решается задача большой размерности. Достижима ли любая требуемая точность результата?

8. Возможно ли равенство $(E - A)^{-1} = E + A + A^2 + A^3 + \ldots + A^k + \ldots$?

Глава 3. ЧИСЛЕННОЕ РЕШЕНИЕ АЛГЕБРАИЧЕСКИХ И ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

В классической математике решение многих задач часто выглядит элементарно. Так при поиске экстремума функции одной переменной предлагается взять ее производную, приравнять к нулю, решить полученное уравнение и т. д. Вне всякого сомнения, что первые два действия в состоянии выполнить любой специалист с незаконченным высшим образованием. Что касается третьего действия, то позвольте усомниться в его элементарности.

Студенту технического вуза, решающему уравнение $\log(x) = 2 - x$ в виде $x = 2 / (\log + 1)$, лучше вспомнить Остапа Бендера с его знаменитой фразой: «Не надо оваций! Графа Монте-Кристо из меня не вышло. Придется переквалифицироваться в управдомы». Выпускник средней школы с легкостью решает квадратные и биквадратные уравнения, простейшие тригонометрические, степенные и другие. Некоторые даже знают о существовании формул Кардано для кубических уравнений. Они понимают тождество понятий «решение уравнения» и «поиск корня уравнения», не ассоциируя этот корень со стоматологией или растениеводством.

В общем случае надежд на простое аналитическое решение достаточно сложного уравнения нет. Более того, доказано, что даже *алгебраическое* (*полиномиальное*) *уравнение* n-й степени с одной переменной

$$P_n(x) \equiv a_0\, x^n + a_1\, x^{n-1} + \ldots + a_{n-1}\, x + a_n = 0$$

при $n > 4$ *неразрешимо в элементарных функциях*. Поэтому решение любого уравнения проводят численно в два этапа (здесь разговор идет лишь о вещественных корнях уравнения).

На первом этапе производится *отделение корней* − поиск интервалов, в которых содержится только по одному корню.

Второй этап решения связан с *уточнением корня* в выбранном интервале (определением значения корня с заданной точностью).

3.1. Отделение корней

В общем случае отделение корней уравнения $f(x) = 0$ базируется на известной теореме, утверждающей, что если непрерывная функция $f(x)$ на концах отрезка $[a, b]$ имеет значения разных знаков, т. е. $f(a) \cdot f(b) < 0$, то на этом отрезке имеется хотя бы один корень. Например, для уравнения $f(x) = x^3 - 6x + 2 = 0$ видим, что при $x \to \infty$ $f(x) > 0$, при $x \to -\infty$ $f(x) < 0$, что уже свидетельствует о наличии хотя бы одного корня. Если же найти точки экстремума из элементарного уравнения $f'(x) = 3x^2 - 6 = 0$ и значения $f(x)$ в этих точках, легко увидеть наличие трех корней (меньшего $-\sqrt{2}$, в интервале от $-\sqrt{2}$ до $\sqrt{2}$ и большего $\sqrt{2}$).

Для уравнения $f(x) = e^x + x = 0$ видим, что $f(-\infty) < 0$, $f(\infty) > 0$. Обнаружив, что $f'(x) = e^x + 1 > 0$ (функция монотонно возрастает), устанавливаем факт наличия единственного корня. Остается найти более короткий подынтервал в диапазоне $(-\infty, \infty)$ (например, $f(x > 0) > 0$ и $f(x = -1) < 0$) и приступить к уточнению.

Если предварительный анализ функции затруднителен, можно «пойти и в лобовую атаку». При уверенности в том, что все корни различны, выбираем некоторый диапазон возможного существования корней (никаких универсальных рецептов не существует) и производим *прогулку* по этому интервалу с некоторым шагом, вычисляя значения $f(x)$ и фиксируя перемены знаков. При выборе шага приходится брать его по возможности большим для минимизации объема вычислений, но достаточно малым, чтобы не пропустить перемену знаков.

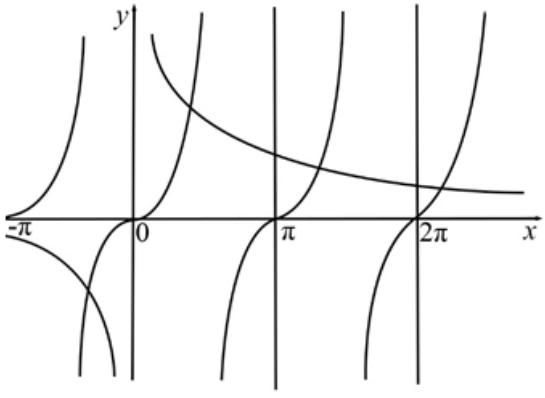


Рис. 3.1

Часто на помощь приходит и графическая интерпретация задачи. Например, для упоминавшегося ранее уравнения $\mathrm{tg}(x) = 1 / x$ можно схематично сделать набросок графиков двух элементарных функций $y = \mathrm{tg}(x)$ и $y = 1 / x$ (рис. 3.1) и убедиться в том, что их пересечение происходит при значениях аргумента из диапазо-

нов $[k\,\pi,\ (k + 0.5)\,\pi]$, $k = 0, \pm1, \pm2, \ldots$ Теперь можно взять интересующий интервал и провести уточнение корня до заданной точности.

Большую сложность представляет случай кратных корней, в окрестности которых отсутствует перемена знаков. При рассмотренном подходе мы не обнаружим кратный корень знакомого квадратного уравнения $x^2 - 2\,x + 1 = 0$. Еще сложнее ситуация с комплексными корнями.

3.2. Оценки корней алгебраических уравнений

Если $f(x)$ − алгебраический многочлен, уравнение называют *алгебраическим* и в противном случае − *трансцендентным*.

Для алгебраических уравнений проблема отделения корней решается несколько проще, равно как проще решается проблема поиска не только действительных, но и комплексных корней. Возьмем алгебраическое уравнение n-й степени

$$P_n(x) \equiv a_0\,x^n + a_1\,x^{n-1} + \ldots + a_{n-1}\,x + a_n = 0 \qquad (3.1)$$

с действительными коэффициентами, причем $a_0 \neq 0$. Согласно основной теореме алгебры, алгебраическое уравнение n-й степени имеет n корней, действительных или комплексно-сопряженных, различных или кратных. Например, уравнение $x^3 - 1 = 0$ имеет один действительный и два комплексно-сопряженных корня, а $x^3 = 0$ − три совпадающих корня (корень кратности 3).

Если Z − корень $P_n(x)$ кратности $m \le n$, т. е. $P_n(x) = (x - Z)^m\,Q_{n-m}(x)$, то $x = Z$ является корнем всех производных $P_n(x)$ до $(m - 1)$-го порядка.

Если уравнение имеет комплексный корень $\alpha + i\,\beta$, то сопряженное число $\alpha - i\,\beta$ будет также корнем той же кратности.

В отличие от трансцендентных, отделение корней алгебраического уравнения опирается на некоторые любопытные оценки, полученные выдающимися математиками XVII–XVIII веков.

Теорема 1. Все корни X_k $(k = 1, 2, \ldots, n)$ полинома в комплексной плоскости лежат в кольце $r < |X_k| < R$, где

$$R = 1 + \frac{A}{|a_0|},\ r = 1/\left(1 + \frac{B}{|a_n|}\right), \qquad (3.2)$$

$A = \max\{|a_1|, |a_2|, \ldots, |a_n|\}$, $B = \max\{|a_0|, |a_1|, \ldots, |a_{n-1}|\}$.

Так для уравнения $P(x) = x^5 - 4x^4 + 5x^3 - 6x + 4 = 0$ с корнями -1, 2, 1, $1 + i$, $1 - i$ (рис. 3.2) получаем

$$A = \max(|-4|, 5, 0, |-6|, 4) = 6,$$
$$B = \max(1, |-4|, 5, 0, |-6|) = 6,$$
$$R = 1 + 6 / 1 = 7,\ r = 1 / (1 + 6 / 4) = 0.4,$$

откуда следует утверждение, что *модули всех корней уравнения лежат в диапазоне* от 0.4 до 7, т. е. в кольце $0.4 < |X_k| < 7$ (рис. 3.3).

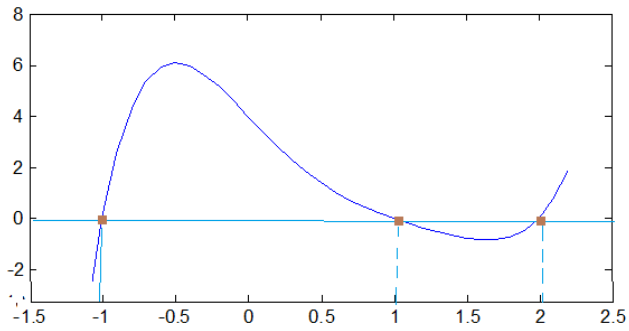


Рис. 3.2

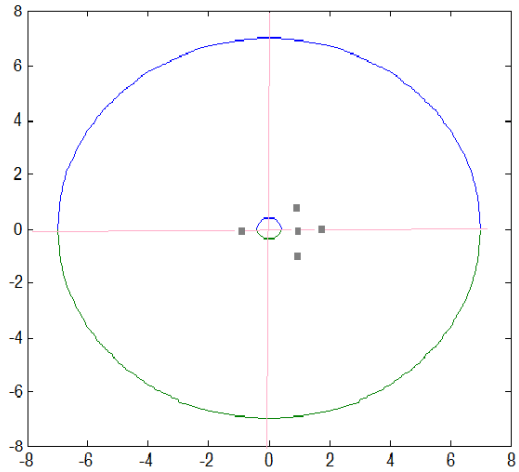


Рис. 3.3

Теорема 2 (теорема Лагранжа*). Пусть $a_0 > 0$ и a_k – первый из отрицательных коэффициентов полинома. Тогда для положительных корней имеет место неравенство

$$x < 1 + \sqrt[k]{\frac{B}{|a_0|}},\ B = \max_{a_i < 0} |a_i|. \qquad (3.3)$$

В рассмотренном выше уравнении отрицателен коэффициент a_1 и верхняя граница положительных корней равна $1 + \max(4, 6) / 1 = 7$.

Для поиска нижней границы отрицательных корней берем

$$P(-x) = -x^5 - 4x^4 - 5x^3 + 6x + 4 = 0,$$

т. е. $x^5 + 4x^4 + 5x^3 - 6x - 4 = 0$. Здесь отрицателен коэффициент a_4 и соответственно

---

* Жозеф Луи Лагранж (1736–1813) – французский математик, астроном и механик, один из великих математиков XVIII века. Сделал механику математической наукой, внес грандиозный вклад в развитие математического анализа, теории чисел, теории вероятностей и численных методов, создал вариационное исчисление.

$$1+\sqrt[4]{\frac{\max(6,4)}{2}}=1+\sqrt[4]{3}\approx 2.3161,$$

т. е. для вещественных корней полинома установлены границы от 2.3161 до 7.

Теорема 3 (теорема Ньютона*). Если при некотором $x = Z$ значения полинома и его производных неотрицательны (n-я производная положительна), то Z может быть принято за верхнюю границу положительных корней полинома.

Возьмем полином $P(-x) = -x^5 - 4x^4 - 5x^3 + 6x + 4$. Для рассматриваемого полинома и его производных, например при $x = 2.5$ $P(x) = x^5 - 4x^4 + 5x^3 - 6x + 4 = 8.53125$, $P'(x) = 5x^4 - 16x^3 + 15x^2 - 6 = 33.0625$, $P''(x) = 20x^3 - 48x^2 + 30x = 87.5$, $P^{(3)}(x) = 60x^2 - 96x + 30 = 165$, $P^{(4)}(x) = 120x - 96 = 204$, $P^{(5)}(x) = 120$. Значения положительны и можно утверждать, что верхняя граница для положительных корней не превышает 2.5.

Если обратиться к уравнению $-P(-x) = -(x^5 + 4x^4 + 5x^3 - 6x - 4) = 0$ и получить для $-P(-x)$ аналогичные оценки при $x = 2.5$, получаем право утверждать, что нижняя граница для отрицательных корней не менее -2.5.

Иногда оказываются полезными и следующие утверждения.

Теорема 4 (теорема Декарта)**. Число положительных корней полинома равно числу перемен знаков в системе его коэффициентов или меньше этого числа на четную величину.

Так для уравнения $P(x) = x^5 - 4x^4 + 5x^3 - 6x + 4 = 0$ (нулевые коэффициенты не учитываются) число положительных корней равно 4 или 2. Если взять $P(-x) = -(x^5 + 4x^4 + 5x^3 - 6x - 4) = 0$, то число перемен знаков равно 1 и исходное уравнение имеет один отрицательный корень.

---

* Исаак Ньютон (1642−1727) − великий английский физик, математик и астроном, один из создателей дифференциального и интегрального исчисления. Сформулировал закон всемирного тяготения и другие основы классической механики.

** Рене Декарт (1596−1650) – французский математик, физик, философ, создатель аналитической геометрии и современной алгебраической символики, механицизма в физике. Всем знакома декартова прямоугольная система координат.

Теорема 5 (теорема Гюа*). Если полином с вещественными коэффициентами имеет только действительные корни, то система его коэффициентов удовлетворяет условию

$$a_k^2 > a_{k-1}\, a_{k+1}\ (k = 1, 2, \ldots, n-1). \qquad (3.4)$$

Обратите внимание, что обратное утверждение неверно. Например, полином $P(x) = x^2 - 4\,x + 5$, где $4^2 > 1 \cdot 5$, обладает двумя комплексными корнями.

3.3. Основные методы уточнения корней уравнения

Прежде чем говорить об уточнении корней, несколько слов об обеспечении точности. Нереально найти *точное* значение корня или добиться обращения функции в нуль. Здесь критерием достижения удовлетворительного результата может служить *абсолютная* или *относительная погрешность корня* (если корень близок к нулю, то лишь относительная погрешность даст необходимое число значащих цифр; если же он весьма велик по абсолютной величине, то критерий абсолютной погрешности дает совершенно излишние верные цифры). Для функций, быстро изменяющихся (осциллирующих) в окрестности корня, привлекателен следующий критерий: *абсолютная величина значения функции* не превышает заданной допустимой погрешности.

3.3.1. Метод дихотомии

Самым простейшим из методов уточнения корней является метод Больцано** *половинного деления* или *метод дихотомии (бисекции)*.

---

* Мальв Жан Поль Гюа (1713−1785) − французский математик. Занимался общей теорией уравнений (определением границ и числа действительных и мнимых корней алгебраического уравнения с геометрической точки зрения), общей теорией кривых высшего порядка.

** Бернард Больцано (1781−1848) − чешский математик, философ и логик. Широко известны его результаты в классическом анализе и теории функций.

Пусть непрерывная функция $f(x)$ на концах отрезка $[a, b]$ имеет значения разных знаков, т. е. $f(a) \cdot f(b) < 0$. Другими словами, на отрезке имеется хотя бы один корень. Возьмем середину отрезка $x = (a + b) / 2$. Если $f(a) \cdot f(x) < 0$, то корень явно принадлежит интервалу от a до $(a + b) / 2$ и в противном случае от $(a + b) / 2$ до b. Берем подходящий из этих интервалов, вычисляем значение функции в его середине и т. д., пока длина очередного интервала не окажется меньше заданной предельной абсолютной погрешности (или выполнятся другие вышеупомянутые критерии). Так как каждое очередное вычисление $f(x)$ сужает интервал поиска вдвое, то при исходном отрезке $[a, b]$ и предельной погрешности ε количество вычислений n определяется условием $(b - a) / 2^n < \varepsilon$, или $n \sim \log_2((b - a) / \varepsilon)$. Например, при исходном единичном интервале и точности порядка шести знаков после десятичной точки достаточно провести лишь 20 вычислений значений функции.

С точки зрения программной реализации метод наиболее прост – в Паскаль-подобной среде этот алгоритм выглядит так:

```
while abs(b-a)>eps do
 begin
  x:=(a+b)/2;
  if f(a)*f(x)>0 then a:=x else b:=x
 end;
```

И он достаточно популярен, хотя существуют и другие более эффективные по затратам времени методы.

3.3.2. Метод хорд

В отличие от метода дихотомии, обращающего внимание лишь на знаки значений функции, но не на сами значения, *метод хорд* использует пропорциональное деление интервала (рис. 3.4). Здесь вычисляются значения функции на концах интервала, и строится «хорда», соединяющая точки $(a, f(a))$ и $(b, f(b))$. Точка пере-

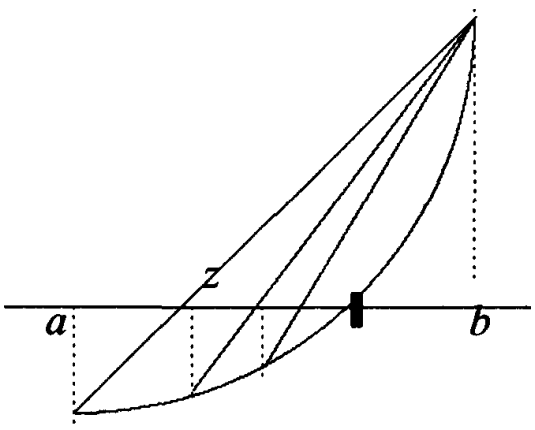


Рис. 3.4

сечения ее с осью абсцисс

$$z = \frac{a\ f(b) - b\ f(a)}{f(b) - f(a)} \tag{3.5}$$

принимается за очередное приближение к корню. Анализируя знак $f(z)$ в сопоставлении со знаком $f(x)$ на концах интервала, сужаем интервал до $[a, z]$ или $[z, b]$ и продолжаем процесс построения хорд до тех пор, пока разница между очередными приближениями не окажется достаточно малой (в пределах допустимой погрешности).

Доказано, что истинная погрешность найденного приближения

$$\left| X^* - Z_n \right| \le \frac{M - m}{m} \left| Z_n - Z_{n-1} \right|, \tag{3.6}$$

где X^* − корень уравнения, Z_n и Z_{n-1} − очередные приближения; m и M − наименьшее и наибольшее значения $|f'(x)|$ на отрезке $[a, b]$.

В случае так называемых быстро осциллирующих функций, где в небольшой окрестности корня значения функции меняются весьма значимо, условие завершения итераций можно взять в виде $|f(x)| < \varepsilon$.

3.3.3. Метод Ньютона – Рафсона (метод касательных)

Другую обширную группу методов уточнения корня представляют итерационные методы, в отличие от методов дихотомии и хорд требующие задания не начального интервала местонахождения корня, а его начального приближения.

Наиболее популярным из таких методов является метод Ньютона − Рафсона (*метод касательных*). Пусть известно некоторое приближенное значение Z_n корня X^*. Применяя формулу Тейлора

$$f(Z_n + h) = f(Z_n) + hf'(Z_n) + \frac{h^2}{2!} f''(Z_n) + \ldots + \frac{h^n}{n!} f^{(n)}(Z_n) + \ldots \tag{3.7}$$

и ограничиваясь в ней двумя членами, имеем

$$f(Z_n + h) \approx f(Z_n) + h\, f'(Z_n) = 0,$$

откуда

$$h = -\frac{f(Z_n)}{f'(Z_n)},\ Z_{n+1} = Z_n - \frac{f(Z_n)}{f'(Z_n)}. \tag{3.8}$$

Геометрически этот метод предлагает построить касательную к кривой $y = f(x)$ в выбранной точке $x = Z_n$, найти точку пересечения ее с осью x и принять эту точку за очередное приближение к корню (рис. 3.5).

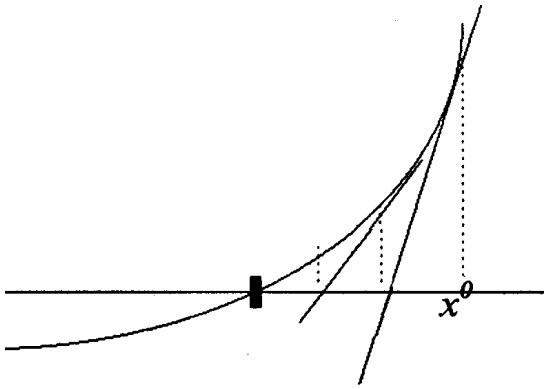


Рис. 3.5

Очевидно, метод касательных обеспечивает сходящийся процесс приближений лишь при выполнении некоторых условий: в зависимости от выбора начального приближения и особенностей $f(x)$ может возникнуть расходящийся процесс (рис. 3.6) или уход к другому корню (рис. 3.7).

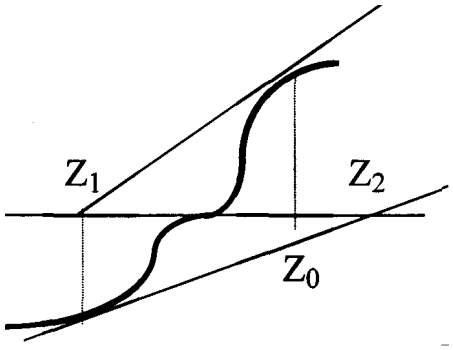


Рис. 3.6

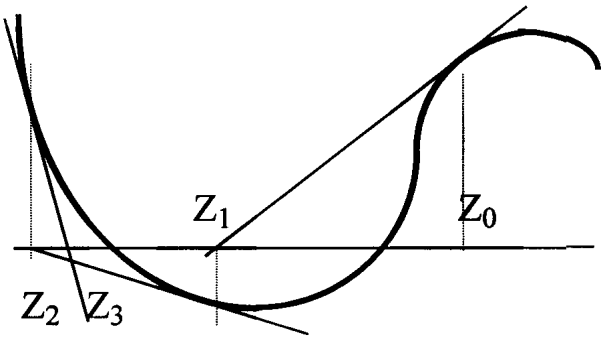


Рис. 3.7

В случае функций, непрерывных на $[a, b]$ вместе со своими производными, для выбора начального приближения полезна следующая теорема [1, 2, 4].

Теорема. Если $f(a)\,f(b) < 0$, $f'(a)$ и $f'(b) \neq 0$ и знакопостоянны на $[a, b]$, то при соблюдении $f(z_0)\,f''(z_0) > 0$, $z_0 \in [a, b]$ итерационный процесс (3.8) сходится (рис. 3.6).

Очевидно, что для функций, производная от которых в окрестности корня близка к нулю, использовать метод Ньютона едва ли разумно.

Вернемся к формуле Тейлора (3.7), учитывая на единицу большее число слагаемых

$$f(Z_n + h) = f(Z_n) + hf'(Z_n) + \frac{h^2}{2!} f''(Z_n) = f(Z_n) + h\left[f'(Z_n) + \frac{h}{2} f''(Z_n) \right] \cong 0.$$

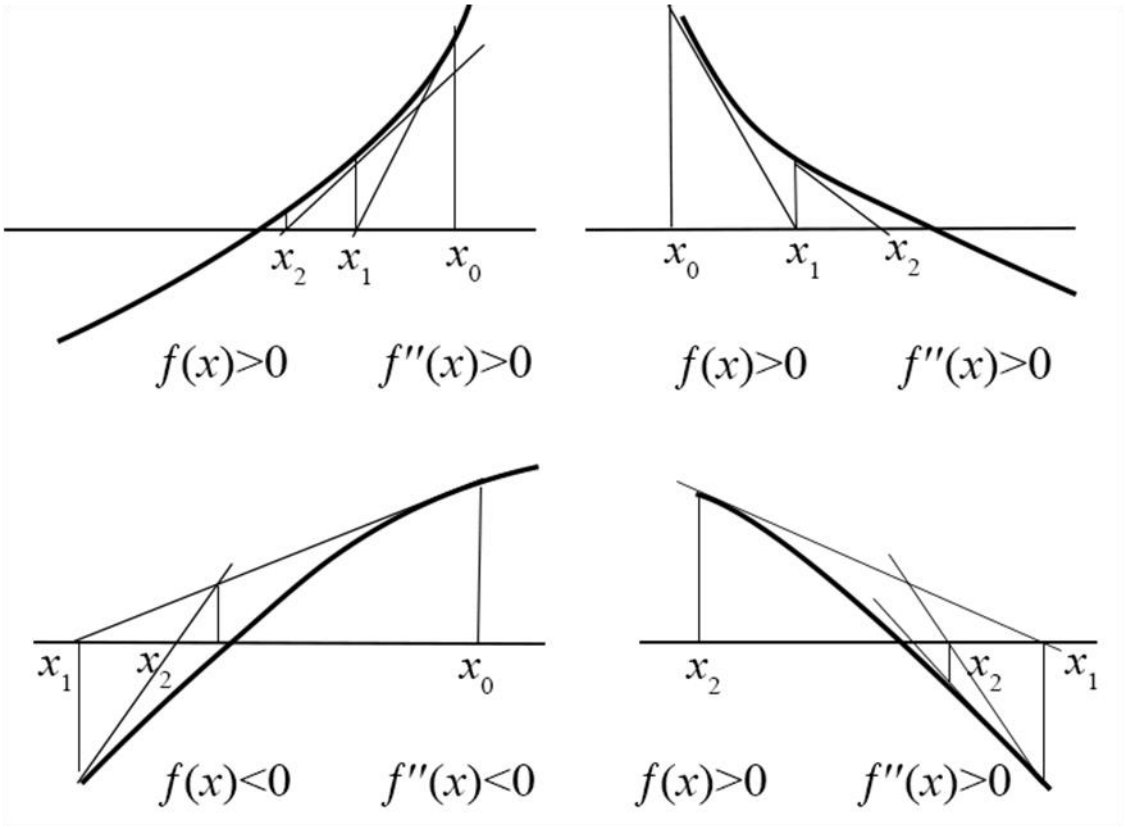


Рис. 3.8

Положив внутри скобок $h = -\dfrac{f(Z_n)}{f'(Z_n)}$, получаем

$$f(Z_n + h) = f(Z_n) + h\left[f'(Z_n) - \frac{1}{2}\frac{f(Z_n)}{f'(Z_n)} f''(Z_n) \right] =$$

$$= f(Z_n) + \frac{1}{2} h \left[\frac{2[f'(Z_n)]^2 - f(Z_n)\, f''Z_n)]}{f'(Z_n)} \right] \cong 0.$$

Откуда получаем формулу

$$h = -\frac{2 f(Z_n) f'(Z_n)}{2[f'(Z_n)]^2 - f(Z_n) \cdot f''(Z_n)} = \frac{\dfrac{f(Z_n)}{f'(Z_n)}}{1 - \dfrac{f(Z_n) f''(Z_n)}{2[f'(Z_n)]^2}}. \qquad (3.9)$$

Для уточнения корня можно воспользоваться *методом Эйткена – Стеффенсена*, который строится как сочетание методов хорд и простой итерации с итерационным процессом вида

$$x_{n+1} = x_n - \frac{f^2(x_n)}{f(x_n) - f(x_n - f(x_n))}, \; n = 0, 1, 2, \ldots, \qquad (3.10)$$

сходящимся при $|x + f(x)| \neq 1$.

Если производная функции мало изменяется в окрестности корня, то можно использовать видоизменение метода

$$Z_{n+1} = Z_n - \frac{f(Z_n)}{f'(Z_0)},\ n = 0, 1, 2, \ldots \qquad (3.11)$$

В случае корня кратности m ($m > 1$) процесс Ньютона существенно замедляется и для его ускорения используют модификацию

$$Z_{n+1} = Z_n - m\ \frac{f(Z_n)}{f'(Z_n)}. \qquad (3.12)$$

Существуют и другие методы уточнения корней, претендующие на более высокую скорость и облегченные условия сходимости.

3.3.4. Метод простой итерации

Для использования этого метода уравнение $f(x) = 0$ приводится к виду $x = \varphi(x)$ и затем строится последовательность значений

$$x_{n+1} = \varphi(x_n),\ n = 0, 1, 2, \ldots \qquad (3.13)$$

Если функция $\varphi(x)$ определена и дифференцируема на некотором интервале, причем $|\varphi'(x)| < 1$, то эта последовательность сходится к корню уравнения $x = \varphi(x)$ на этом интервале. Геометрическая интерпретация процесса представлена на рис. 3.9–3.11. Здесь первые два примера демонстрируют одностороннее и двустороннее приближение к корню, третий же выступает иллюстрацией расходящегося процесса ($|\varphi'(x)| > 1$).

Если в окрестности корня $f'(x) > 0$, то равносильное уравнение можно взять в виде $x = x - \lambda f(x)$, т. е. $\varphi(x) = x - \lambda f(x)$, где $\lambda > 0$ подбирается таким образом, чтобы в окрестности корня выполнялось условие $0 < \varphi'(x) = 1 - \lambda f'(x) \le 1$. Соответственно может быть построен итерационный процесс

$$x_{n+1} = x_n - \frac{f(x_n)}{M},\ n = 0, 1, 2, \ldots, \qquad (3.14)$$

где $M \ge \max|f'(x)|$ (в случае $f'(x) < 0$ возьмите функцию $f(x)$ с противоположным знаком).

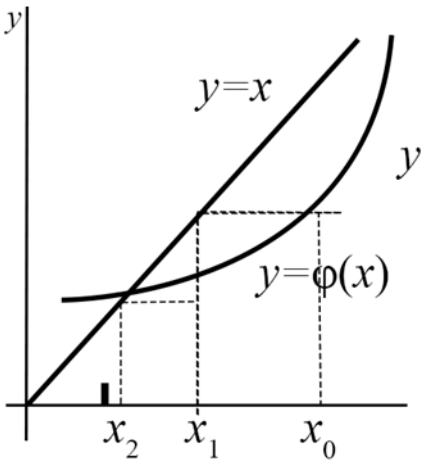


Рис. 3.9

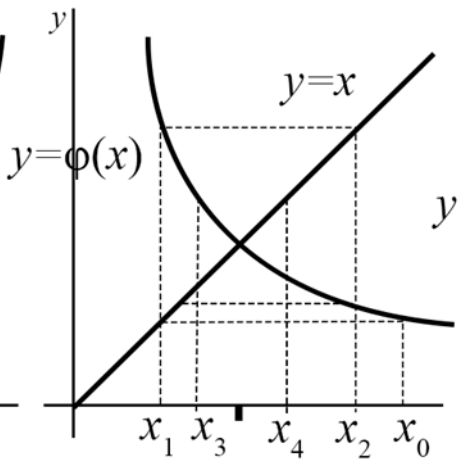


Рис. 3.10

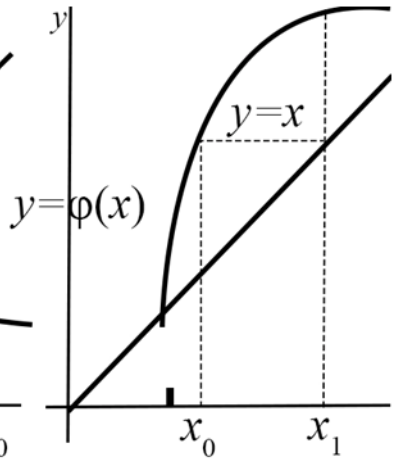


Рис. 3.11

Возьмем для примера уравнение $x^3 + x - 1000 = 0$. Очевидно, что его корень будет порядка 10. Переписав уравнение в виде $x = 1000 - x^3$ и начав итерационный процесс при $x_0 = 10$, из первых же приближений обнаруживаем отсутствие сходимости: 10, 0, 1000, −999999000, … Если же учесть $f'(x) = 3x^2 + 1 > 0$ и принять за приближенное значение максимума производной $M = 300$, то можно построить сходящийся итерационный процесс на основе представления $x = x - \frac{x^3 + x - 1000}{300}$ с последовательностью приближений 10, 9.9666666666, 9.9666667901, 9.9666667905, …

3.3.5. Кубические уравнения. Век нынешний и век минувший

Интерес к решению кубических уравнений возник тысячелетия назад. Уже в III веке древнегреческие математики проявляли интерес к *задаче удвоения куба* (поиск длины стороны куба с объемом вдвое большим имеющегося куба).

Возьмем уравнение $P(x) = x^3 + 3x^2 + 10x - 40 = 0$ и подвергнем его небольшому анализу. Воспользуемся ранее приведенными сведениями о корнях многочленов и методе Ньютона. Наряду с тривиальной информацией о числе корней, согласно теореме Декарта, по числу перемен знака обнаруживаем наличие одного положительного и двух или не одного неотрицательного корня. Первый отрицательный коэффициент a_k соответствует третьему месту ($k = 3$) и, по теореме Лагранжа, для положительных корней имеет место неравенство

$$B = \max_{a_i < 0} |a_i| = 40,\ x < 1 + k\sqrt{\frac{B}{|a_0|}} = 1 + \sqrt[3]{40} \approx 4.42.$$

Наконец, по теореме о кольце, все корни полинома в комплексной плоскости лежат в кольце $r < |X| < R$, где с учетом

$A = \max\{|a_1|, |a_2|, |a_3|\} = \max(3, 10, 40) = 40,$

$B = \max\{|a_0|, |a_1|, |a_2|\} = \max(1, 3, 10) = 10,$

$$R = 1 + \frac{A}{|a_0|} = 1 + 40 = 41,\ r = 1/(1 + \frac{B}{|a_n|}) = 1/(1 + \frac{10}{|40|}) = 1.25.$$

В итоге обнаруживаем, что искомый положительный корень лежит в диапазоне от 1.25 до 4.42.

Решив проблему выбора начального приближения, обращаемся к обычному методу Ньютона

$$x_{k+1} = x_k - \frac{x_k^3 + 3x_k^2 + 10x_k - 40}{3x_k^2 + 6x_k + 10}$$

и, выбрав $x_0 = 3$, получаем последовательность оценок 2.2000, 2.0100, 2.0000, 2.0000. При $x_0 = 1.25$ – последовательность 2.1901, 2.0090, 2.0000. Даже при удаленном выборе $x_0 = 40$ потребовалось 8 итераций 26.3050, 17.1679, 7.0312, 4.4066, 2.8449, 2.1486, 2.0056, 2.0000. Разделив $P(x)$ на $x - 2$, получаем квадратный трехчлен $x^2 + 5x + 20$, корни которого $x_{2,3} = -2.5 \pm i\sqrt{13.75}$.

Решение этой задачи практически мгновенно может быть получено при наличии персонального компьютера и хорошего программного обеспечения. А как быть, если такого компьютера «нет в кустах»?

Еще 65 лет назад отсутствовало серийное производство ЭВМ. Имелись уникальные образцы в США, СССР и Англии, серийное производство в США началось в 1952 году с ЭВМ IBM 701, а в СССР – в 1954 году с шедевра той эпохи ЭВМ «Стрела». И долгие годы мир взирал на программистов как на чудотворцев.

Программа средних школ по математике предусматривала знакомство с ныне практически забытыми формулами Кардано*, предназначенными для решения кубических уравнений.

Обратимся к решению *кубического уравнения*

$$x^3 + a\,x^2 + b\,x + c = 0, \tag{3.15}$$

где a, b, c – вещественные числа. Проведем замену переменной x

$$x = y - \frac{a}{3}. \tag{3.16}$$

Подставив (3.16) в (3.15), получаем уравнение

$$y^3 + p\,y + q = 0, \tag{3.17}$$

где

$$p = b - \frac{a^2}{3},\ q = c + \frac{2a^3}{27} - \frac{ab}{3}. \tag{3.18}$$

На последующем этапе сведения (3.17) к решению квадратного уравнения производим замену переменной

$$y = z - \frac{p}{3z}, \tag{3.19}$$

получая

$$z^3 - \frac{p^3}{27z^3} + q = 0. \tag{3.20}$$

Откуда без особых затруднений при $\Delta = \frac{q^2}{4} + \frac{p^3}{27} > 0$ находим

$$z_{1,2}^3 = -\frac{q}{2} \pm \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}$$

и

$$z_{1,2} = \sqrt[3]{-\frac{q}{2} \pm \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}}\ . \tag{3.21}$$

Подстановкой любого из найденных значений в (3.19) можно убедиться в тождественности получаемых значений y и, с помощью (3.16), найти один из корней x уравнения (3.15).

---

* Джироламо Кардано (1501–1576) – итальянский математик, инженер. Считается изобретателем карданного вала. Автор «Книги об игре в кости» – одного из первых значимых трудов по комбинаторике. Ему принадлежит первая публикация формул для решения кубических уравнений, неизвестного математикам того времени.

Не записывая формулу Кардано в громоздком развернутом виде, обратимся к приведенному ранее уравнению $P(x) = x^3 + 3\,x^2 + 10\,x - 40 = 0$. Имеем $a = 3,\ b = 10,\ c = -40$. Выполнив замену (3.16) $x = y - 1$, получим

$$p = b - \frac{a^2}{3} = 10 - \frac{3^2}{3} = 7,\ q = c + \frac{2a^3}{27} - \frac{ab}{3} = -40 + \frac{2\cdot 3^3}{27} - \frac{3\cdot 10}{3} = -48.$$

Далее приступаем к решению уравнения $y^3 + 7\,y - 48 = 0$.

Заменой $y = z - \frac{7}{3z}$ приходим к уравнению $z^3 - \frac{7^3}{27z^3} - 48 = 0$,

откуда один из корней равен

$$z^3 = \frac{48}{2} + \sqrt{\frac{48^2}{4} + \frac{7^3}{27}} = 24 + \sqrt{\frac{24^2}{1} + \frac{7^3}{27}} \approx 48.2632.$$

Отыскав $z = z^* = 3.6409^*$, получаем $y = z - 7 / (3\,z) = 3.0000$ и $x = 2$.

Можно показать [16], что в ситуации равенства нулю дискриминанта $\Delta = \frac{q^2}{4} + \frac{p^3}{27} = 0$ имеем $y = -\frac{3q}{p}$.

Гораздо труднее было искать решение при $\Delta = \frac{q^2}{4} + \frac{p^3}{27} < 0$ в то время, когда право на существование комплексных чисел еще обсуждалось и не было знаменитой формулы Эйлера

$$A + B\,i = R\cdot e^{i\varphi} = R\cdot(\cos\varphi + i\cdot\sin\varphi),$$

где

$$R = \sqrt{A^2 + B^2},\ \varphi = \operatorname{arctg}\left(\frac{B}{A}\right).$$

(да и сегодня не всякий школьник осилит приведенную ниже процедуру поиска).

В нашем случае, отыскав z^3, получаем одно из значений:

$$z = \sqrt[3]{-\frac{q}{2} + i\sqrt{-(\frac{q}{2})^2 - (\frac{p}{3})^3}}.$$

Для поиска кубического корня обозначим

$$A = -\frac{q}{2},\ B = \sqrt{-(\frac{q}{2})^2 - (\frac{p}{3})^3}.$$

---

* Согласно теореме Виета о сумме и произведении корней полинома, наряду с z^*, существуют корни $z = (-\frac{1}{2} \pm \frac{\sqrt{3}}{2})\cdot z^*$.

Отыскав соответствующие

$$R = \sqrt{-(p/3)^3}\,,\ \varphi = \operatorname{arctg}\left[\sqrt{\left|-(\frac{q}{2})^2 - (\frac{p}{3})^3\right|} / (-\frac{q}{2})\right], \qquad (3.22)$$

находим $z = \sqrt[3]{Re^{i\varphi}} = \sqrt[3]{R}e^{i\frac{\varphi}{3}}$ и из (3.19) получаем

$$x = 2\frac{p}{3}\cos\left(\frac{\varphi}{3}\right) - \frac{a}{3}\cdot \qquad (3.23)$$

Существует третий любопытный путь решения задачи, если вместо (3.19) взять замену

$$y = -\frac{q}{p}z\,. \qquad (3.24)$$

Тогда возникает уравнение

$$z^3 + A\,z - A = 0,\ A = p^3 / q^2. \qquad (3.25)$$

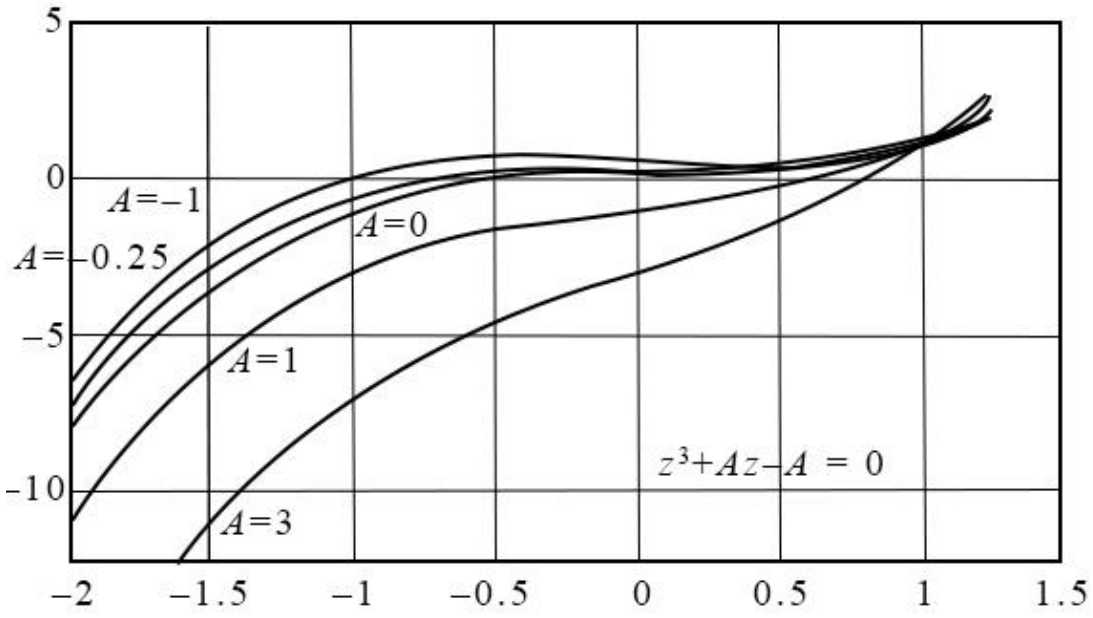


Рис. 3.12

Если сделать графический набросок поведения $f(z, A) = z^3 + A\,z - A$ (рис. 3.12) и выполнить некоторые оценки, то можно обнаружить диапазоны местоположения искомых корней: при $A > 0$ $0 < z < 1$ (с ростом A приближается к 1, с уменьшением – к 0); при $A = 0$ $z = 0$; при $A = -0.5$ $z = -1$; при $A < -0.5$ $z < -1$. Уже эта информация дает возможность поиска корня (3.25) хотя бы методом дихотомии или хорд.

При выборе метода полезно учесть, что $f''(z) = 3z$ и при $z > 0$ $f(z)$ выпуклая и в противном случае вогнутая (рис. 3.13).

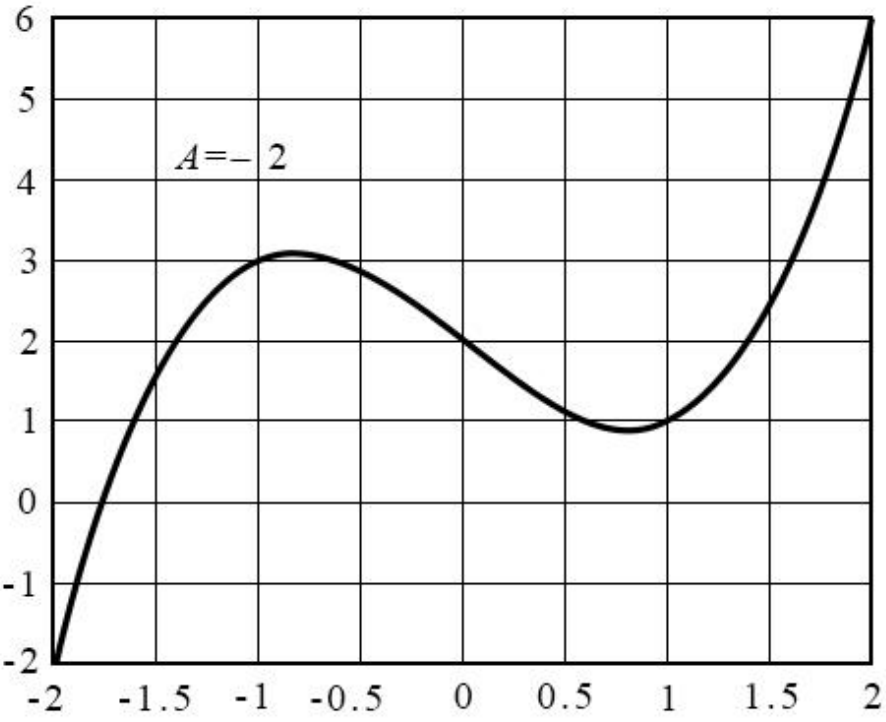


Рис. 3.13

Для $A > 1$ можно прибегнуть к простой итерации $z_{k+1} = 1 - z_k^3 / A$, обнаруживая достаточно быстро сходящийся процесс. Так при $A = 10$ получаем последовательность 0.9000, 0.9271, 0.9203, 0.9221, 0.9216, 0.9217, 0.9217.

Что касается малых A, то есть резон взять $z_{k+1} = \sqrt[3]{A(1 - z_k^3)}$ и при $A = 0.5$ выбрать $x_0 = 0$, получая последовательность 0.7937, 0.6300, 0.7211, 0.6786, 0.7005, 0.6897, 0.6952, 0.6925, 0.6938, 0.6931, 0.6935, 0.6933, 0.6933.

При $A = -1 < -0.5$ воспользуемся методом Ньютона $z_{k+1} = z_k - \frac{z_k^3 + Az_k - A}{3z_k^2 + A}$ и получим последовательность приближений –1.1000, –1.3924, –1.3286, –1.3247, –1.3247.

Таким образом для кубического уравнения можно сочетать приемы Кардано сведения (3.15) к форме (3.25) с последующей итерационной процедурой вместо использования многостраничных таблиц [16].

3.3.6. Метод наискорейшего спуска

Популярный в приложениях *метод наискорейшего спуска* для случая одной переменной упрощается до элементарнейшего алгоритма.

Выбираем приближение к корню X_n и начальный шаг H. Если $f(X_n) > 0$, производим переход в точку X_{n+1} слева или справа от X_n в зависимости от знака $f'(X_n)$ (в сторону убывания функции):

$$X_{n+1} = X_n - \frac{f'(X_n)}{|f'(X_n)|} H. \qquad (3.26)$$

Если $f(X_{n+1}) < f(X_n)$, то эта точка принимается за очередное приближение и процесс продолжается; в противном случае возвращаемся в точку X_n, уменьшаем шаг вдвое и повторяем переход. Процесс продолжается до малого шага или малого значения функции (его сходимость зависит от близости начального приближения к корню и удачного выбора шага).

Очевидно, что такой метод приемлем и в случае кратных корней.

3.3.7. Обобщенный метод Ньютона (поиск комплексных корней)

Этот метод является, как нам кажется, самым эффективным методом поиска любых корней уравнения и, в частности, всех корней алгебраического многочлена [1].

Возьмем уравнение $f(Z) = 0$, выберем начальное приближение к корню $Z_k = X_k + i\,Y_k$ и начальный шаг t (например, $t = 1$). Находим первую отличную от нуля производную $f^{(m)}(Z_k)$ и последующий процесс приближений ведем по формуле

$$Z_{k+1} = Z_k + t\left(-\frac{f(Z_k)}{f^{(m)}(Z_k)}\right)^{\frac{1}{m}}. \qquad (3.27)$$

Если $|f(Z_{k+1})| > |f(Z_k)|$, то t уменьшаем вдвое (как в методе наискорейшего спуска) и повторяем переход до $|f(Z_k)| < \varepsilon$ или близости очередных приближений.

В случае, когда $f(Z)$ является алгебраическим многочленом (полиномом), метод сходится при любом начальном приближе-

нии к одному из корней. Для аналогичного поиска других корней полинома достаточно его степень понизить делением на $(Z - Z^*)$, где Z^* − найденный корень.

Для примера найдем корни алгебраического уравнения $P(Z) = Z^3 - 5Z^2 + 9Z - 5 = 0$. Пусть $t = 1$ и начальное приближение $Z_0 = 0$. Отыскав первую производную $P'(Z) = 3Z^2 - 10Z + 9$ и обнаружив $P'(Z_0) = 9 \neq 0$, ищем первое приближение ($m = 1$):

$$Z_1 = Z_0 + 1\left(-\frac{P(Z_0)}{P'(Z_0)}\right) = 0 + 1\left(-\frac{-5}{9}\right) \approx 0.556.$$

Значение полинома уменьшилось, т. к.
$|P(Z_1) \approx -1.37| < |P(Z_0) = 5|$, продолжаем итерации:

$$Z_2 = Z_2 + 1\left(-\frac{P(Z_1)}{P'(Z_1)}\right) \approx 0.556 + 1\left(-\frac{-1.37}{2.5}\right) = 1.004,$$

$$Z_3 = Z_2 + 1\left(-\frac{P(Z_2)}{P'(Z_2)}\right) \approx 1.004 + 1\left(-\frac{0.008}{2}\right) = 1.000.$$

В итоге обнаруживаем, что $Z = 1$ − один из корней уравнения. Разделив $P(Z)$ на $Z - 1$, получаем уравнение $P_2(Z) = Z^2 - 4Z + 5 = 0$ с производной $P_2'(Z) = 2Z - 4$. Вновь возьмем $t = 1$ и начальное приближение $Z_0 = 0$. Аналогично предыдущему рассмотрению, имеем:

$P_2(Z_0) = \mathbf{5}$; $P_2'(Z_0) = 2 \cdot 0 - 4 = -4$; $Z_1 = 0 + 1 \cdot 5 / 4 = 1.25$;

$P_2(Z_1) = \mathbf{1.5625}$; $P_2'(Z_1) = -1.5$; $Z_2 = 1.25 + 1 \cdot 1.5625 / 1.5 \approx 2.2916$;

$P_2(Z_2) = \mathbf{1.0851}$; $P_2'(Z_2) = 0.5833$; $Z_3 = 2.2916 - 1 \cdot 1.0851 / 0.5833 \approx$ ≈ 0.4315;

$P_2(Z_3) = \mathbf{3.4600}$; (значение полинома > $P_2(Z_2) = \mathbf{1.0851}$ − уменьшаем t до 0.5); $Z_4 = 2.2916 - 0.5 \cdot 1.0851 / 0.5833 \approx 1.3616$;

$P_2(Z_4) = \mathbf{1.4075}$; (значение полинома > $P_2(Z_2) = \mathbf{1.0851}$ − уменьшаем t до 0.25); $Z_5 = 2.2916 - 0.25 \cdot 1.0851 / 0.5833 \approx 1.8266$;

$P_2(Z_5) = \mathbf{1.0301}$; $P_2'(Z_5) = -0.3467$;

$Z_6 = 1.8266 + 0.25 \cdot 1.0301 / 0.3467 \approx 2.5693$;

$P_2(Z_6) = \mathbf{1.3241}$; (значение полинома > $P_2(Z_5) = \mathbf{1.0301}$; следовательно, уменьшаем t до 0.125 и т. д.).

После серии переходов с многократным изменением t получается оценка $\boldsymbol{Z_{20}} = 2.0000$ со значениями $P_2(Z_{20}) = \mathbf{1}$; $P_2'(Z_{20}) = \mathbf{0}$. Поскольку значение производной обратилось в нуль, берем $m = 2$, находим ненулевую $P_2''(Z_{20}) = 2$ и получаем (восстановив $t = 1$)

$$Z_{21} = Z_{20} + 1\left(-\frac{P_2(Z_{20})}{P_2''(Z_{20})}\right)^{\frac{1}{2}} \approx 2.0000 + 1\left(-\frac{1.00}{2}\right)^{\frac{1}{2}} = 2.0000 + \frac{1}{\sqrt{2}}i.$$

Здесь $P_2(Z_{21}) = \mathbf{0.5}$; $P_2'(Z_{21}) = i / \sqrt{2}$;
$Z_{22} = (2.0000 + i / \sqrt{2}) - 0.5 / (i / \sqrt{2}) = 2 + i\sqrt{2}$;
$P_2(Z_{22}) = \mathbf{3} > P_2(Z_{21})$ (вновь уменьшаем t до 0.5 и т. д. до получения оценки, близкой к истинному корню $\boldsymbol{Z = 2 + i}$. В силу вещественности коэффициентов уравнения третий из корней является комплексно-сопряженным $\boldsymbol{Z = 2 - i}$.

Очевидно, что использование этого метода при ручных вычислениях едва ли реально (существуют прекрасные его программные реализации в различных языковых средах). Примером удачной такой реализации служит одна из функций MatLab `roots(P)` – поиск всех корней многочлена.

Так для уравнения $2x^5 - 100x^2 + 2x - 1 = 0$ оператор `roots([2 0 0 -100 2 -1])` выдаст все 5 корней:
`-1.8491 ± 3.1897i, 3.6783, 0.0100 ± 0.0995i`.

Кстати, в этой библиотеке есть удобные процедуры поиска коэффициентов полинома по заданным корням `poly([список корней])` и вычисления значений полинома при заданном списке значений аргумента `polyval ([коэффициенты полинома], [аргументы])`.

3.3.8. Коротко о других методах

Кроме метода Ньютона для поиска всех корней многочлена можно рекомендовать и ряд других методов [16].

Метод скорейшего спуска. Здесь многочлен $P_m(z)$ представляется в виде $P_m(z) = u(x, y) + i \cdot v(x, y)$, где $z = x + i \cdot y$, и ставится задача минимизации до нуля функции $u^2 + v^2$ последовательным спуском в направлении, обратном градиенту.

Метод парабол. Этот метод основывается на идеях последовательного построения интерполяционного многочлена Лагранжа второй степени по выбранным трем точкам комплексной плоскости, поиске его корня и замене одной из указанных точек найденным корнем.

Метод Мюллера [6, с. 312]. За исключением некоторых деталей этот метод совпадает с методом парабол. Описание алгоритма (без традиционных опечаток) приведено в [13].

Метод Лобачевского – Греффе [4, с. 176]. Это один из самых популярных в свое время (в докомпьютерную эпоху) методов поиска корней многочлена, основанный на идее квадрирования корней. Для компьютерной реализации неудобен из-за сложностей представления чисел с большими степенями и значительной потери точности.

Метод Хичкока. Аналогично методу парабол в методе Хичкока используется идея выделения квадратного множителя. Отыскивается приближенная оценка пары его корней и последовательно уточняются коэффициенты указанного множителя.

3.4. Решение систем нелинейных уравнений

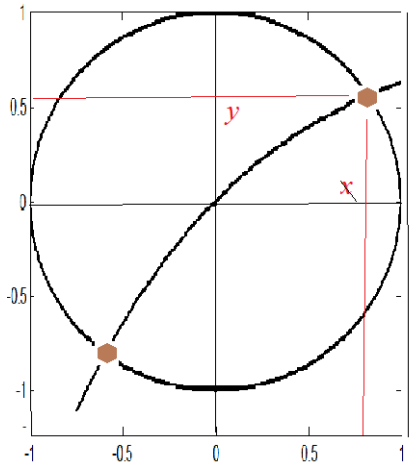


Рис. 3.14

Рассмотрим для простоты систему лишь двух уравнений

$$F_1(x, y) = 0,\ F_2(x, y) = 0. \qquad (3.28)$$

Если преобразовать ее к виду

$$x = \Phi_1(x, y),\ y = \Phi_2(x, y) \qquad (3.29)$$

и задаться каким-то начальным приближением, то можно построить *процесс простой итерации*

$$x_k = \Phi_1(x_{k-1}, y_{k-1}),\ y_k = \Phi_2(x_{k-1}, y_{k-1}), \qquad (3.30)$$

сходящийся, если в окрестности корня, содержащей начальное приближение, выполняются условия

$$\left|\frac{\partial\Phi_1}{\partial x}\right| + \left|\frac{\partial\Phi_2}{\partial x}\right| \le q_1 < 1,\ \left|\frac{\partial\Phi_1}{\partial y}\right| + \left|\frac{\partial\Phi_2}{\partial y}\right| \le q_2 < 1. \qquad (3.31)$$

Вместо простой итерации можно использовать *метод Зейделя*

$$x_k = \Phi_1(x_{k-1}, y_{k-1}),\ y_k = \Phi_2(x_k, y_{k-1}) \qquad (3.32)$$

с более высокой сходимостью.

Возьмем для примера систему из двух уравнений $x^2 + y^2 = 1$, $e^{-x} + y = 1$, преобразуем ее к виду $x = \sqrt{1 - y^2}$, $y = 1 - e^{-x}$ и по графическому наброску (рис. 3.14) выбираем начальное приближение. Метод Зейделя дает оценки

k	0	1	2	3	4	5	6	7	8
x_k	0.5	0.866	0.815	0.830	0.826	0.8270	0.8267	0.8268	0.8267
y_k	0.5	0.579	0.557	0.564	0.562	0.5627	0.5625	0.5626	0.5625

(на 20-й итерации достигается «точное» решение $x = 0.82677005520$, $y = 0.56254002155$).

Можно использовать для решения системы и *метод Ньютона*. Выберем начальное приближение (x_k, y_k). Считая истинное решение равным $x = x_k + h$, $y = y_k + t$, разложим функции в ряд Тейлора и ограничимся линейными членами разложения:

$$\begin{aligned} F_1(x, y) &\cong F_1(x_k, y_k) + h\frac{\partial F_1(x_k, y_k)}{\partial x} + t\frac{\partial F_1(x_k, y_k)}{\partial y} = 0, \\ F_2(x, y) &\cong F_2(x_k, y_k) + h\frac{\partial F_2(x_k, y_k)}{\partial x} + t\frac{\partial F_2(x_k, y_k)}{\partial y} = 0. \end{aligned} \qquad (3.33)$$

Если определитель этой системы отличен от нуля

$$J(x_k, y_k) = \begin{vmatrix} \dfrac{\partial F_1(x_k, y_k)}{\partial x} & \dfrac{\partial F_1(x_k, y_k)}{\partial y} \\ \dfrac{\partial F_2(x_k, y_k)}{\partial x} & \dfrac{\partial F_2(x_k, y_k)}{\partial y} \end{vmatrix} \neq 0, \qquad (3.34)$$

то очередное приближение сводится к поиску (правило Крамера)

$$\begin{aligned} x_{k+1} &= x_k - \begin{vmatrix} F_1 & \dfrac{\partial F_1}{\partial y} \\ F_2 & \dfrac{\partial F_2}{\partial y} \end{vmatrix} / J(x_k, y_k), \\ y_{k+1} &= y_k - \begin{vmatrix} \dfrac{\partial F_1}{\partial x} & F_1 \\ \dfrac{\partial F_2}{\partial x} & F_2 \end{vmatrix} / J(x_k, y_k). \end{aligned} \qquad (3.35)$$

Обратимся к системе из n уравнений

$$F_i(X) = 0,\ i = 1, 2, \ldots, n, \qquad (3.36)$$

где $X = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ (вектор-столбец). Пусть $X = X^{(k)} + \varepsilon^{(k)}$. Разлагая вектор-функцию $F(X) = [f_1(X), f_2(X), \ldots, f_n(X)]^{\mathrm{T}}$ в ряд Тейлора с учетом лишь линейных членов, получаем систему линейных уравнений

$$f_i\left(X^{(k)}\right) + \frac{\partial F_i\left(X^{(k)}\right)}{\partial x_1}\varepsilon_1^{(k)} + \frac{\partial F_i\left(X^{(k)}\right)}{\partial x_2}\varepsilon_2^{(k)} + \ldots + \frac{\partial F_i\left(X^{(k)}\right)}{\partial x_n}\varepsilon_n^{(k)} = 0,$$

где $i = 1, 2, \ldots, n$. Если обозначить через $W(X)$ матрицу частных производных $F(X)$ (такую матрицу обычно называют *якобианом* функции), то эта система в матричном виде представится как $F(X^{(k)}) + W(X^{(k)})\ \varepsilon^{(k)} = 0$. Откуда в представлении через обратную матрицу (естественно, что ее определитель должен быть ненулевым) получится очередная приближенная оценка

$$X^{(k+1)} = X^{(k)} - W^{-1}(X^{(k)})\ F(X^{(k)}) = 0,\ k \geq 0. \qquad (3.37)$$

Для примера возьмем систему

$$x^2 + y^2 + z^2 = 1,$$
$$2x^2 + y^2 - 4z = 0,$$
$$3x^2 - 4y + z^2 = 0.$$

Обозначив

$$X = \begin{vmatrix} x \\ y \\ z \end{vmatrix},\quad F(X) = \begin{vmatrix} x^2 + y^2 + z^2 - 1 \\ 2x^2 + y^2 - 4z \\ 3x^2 - 4y + z^2 \end{vmatrix},\quad W(X) = \begin{vmatrix} 2x & 2y & 2z \\ 4x & 2y & -4 \\ 6x & -4 & 2z \end{vmatrix}$$

и выбрав начальное приближение, имеем

$$X^0 = \begin{vmatrix} 0.5 \\ 0.5 \\ 0.5 \end{vmatrix},\quad F(X^0) = \begin{vmatrix} -0.25 \\ -1.25 \\ -1.00 \end{vmatrix},\quad W(X^0) = \begin{vmatrix} 1 & 1 & 1 \\ 2 & 1 & -4 \\ 3 & -4 & 1 \end{vmatrix}.$$

Отыскав

$$W^{-1}(X^0) = \begin{vmatrix} 3/8 & 1/8 & 1/8 \\ 7/20 & 1/20 & -3/20 \\ 11/40 & -7/40 & 1/40 \end{vmatrix},$$

получаем

$$X^1 = \begin{vmatrix} 0.5 \\ 0.5 \\ 0.5 \end{vmatrix} - \begin{vmatrix} 3/8 & 1/8 & 1/8 \\ 7/20 & 1/20 & -3/20 \\ 11/40 & -7/40 & 1/40 \end{vmatrix} \cdot \begin{vmatrix} -0.25 \\ -1.25 \\ -1.00 \end{vmatrix} = \begin{vmatrix} 0.875 \\ 0.500 \\ 0.375 \end{vmatrix},\ F(X^1) = \begin{vmatrix} 0.15625 \\ 0.28125 \\ 0.43750 \end{vmatrix}.$$

Аналогично получаем

$$X^2 = \begin{vmatrix} 0.78981 \\ 0.49662 \\ 0.36993 \end{vmatrix}, \qquad X^3 = \begin{vmatrix} 0.78521 \\ 0.49662 \\ 0.36992 \end{vmatrix}, \qquad F(X^3) = \begin{vmatrix} 0.00001 \\ 0.00004 \\ 0.00005 \end{vmatrix}.$$

Разумеется, этот метод, как и метод простой итерации, условно сходящийся.

Как и в случае систем линейных уравнений, задачу решения системы (3.36) можно заменить задачей *минимизации до нуля* значений функции

$$\Phi(X) = \sum_{i=1}^{n} A_i F_i^2(X), \tag{3.38}$$

где $A_i > 0$ − весовые коэффициенты.

Решению систем нелинейных уравнений посвящена обширная литература, существует великое множество разнообразных итерационных методов, использование которых реально лишь при наличии компьютера. В случае реальных задач большой размерности априори установить факт сходимости невозможно, нужен компьютерный эксперимент, и трудности решения могут оказаться исключительными.

Глава 4. ПРОБЛЕМА СОБСТВЕННЫХ ЗНАЧЕНИЙ И ЕЕ РЕШЕНИЯ

Проблема собственных значений возникает при решении многих задач, связанных с колебательными процессами, анализом устойчивости систем, оценкой сходимости итерационных процессов и др. Она состоит в поиске значений параметра λ, при которых, наряду с нулевым, существуют и *ненулевые решения* системы линейных уравнений $(A - \lambda B) X = 0$, где A и B − квадратные матрицы n-го порядка.

В случае равенства $B = E$ (ниже мы ограничимся рассмотрением только этого случая) λ называют *собственными числами* (*собственными значениями*) матрицы A и упомянутые решения − *собственными векторами* матрицы.

4.1. Собственные числа и векторы

Очевидно, что система уравнений

$$\begin{aligned}
&(a_{11} - \lambda)\, x_1 + a_{12}\, x_2 + \ldots + a_{1n}\, x_n = 0,\\
&a_{21}\, x_1 + (a_{22} - \lambda)\, x_2 + \ldots + a_{2n}\, x_n = 0,\\
&\ldots\ldots\ldots\ldots\ldots\\
&a_{n1}\, x_1 + a_{n2}\, x_2 + \ldots + (a_{nn} - \lambda)\, x_n = 0
\end{aligned} \qquad (4.1)$$

имеет ненулевые решения лишь в случае, если определитель ее матрицы коэффициентов обращается в нуль, т. е.

$$\left|A - \lambda\ E\right| = 0. \qquad (4.2)$$

Вычислив определитель, получаем уравнение n-й степени (это уравнение называют *характеристическим*, или *вековым*):

$$\lambda^n - p_1\lambda^{n-1} + p_2\lambda^{n-2} - \ldots + (-1)^{n-1} p_{n-1}\lambda + (-1)^n p_n = 0, \qquad (4.3)$$

где

$$p_1 = \sum_{i=1}^{n} a_{ii},\ p_2 = \sum_{i<k} \begin{vmatrix} a_{ii} & a_{ik} \\ a_{ki} & a_{kk} \end{vmatrix},\ p_3 = \sum_{i<k<m} \begin{vmatrix} a_{ii} & a_{ik} & a_{im} \\ a_{ki} & a_{kk} & a_{km} \\ a_{mi} & a_{mk} & a_{mm} \end{vmatrix}, \ldots,\ p_n = |A|. \qquad (4.4)$$

Очевидно, что оно имеет n корней, действительных или комплексно-сопряженных, совпадающих или различных. Так для

$$A = \begin{vmatrix} 5 & -2 & 1 \\ 3 & -2 & 3 \\ 3 & -6 & 7 \end{vmatrix}$$

посредством (4.4) получаем коэффициенты уравнения

$$p_1 = 5 - 2 + 7 = 10,$$

$$p_2 = \begin{vmatrix} 5 & -2 \\ 3 & -2 \end{vmatrix} + \begin{vmatrix} 5 & 1 \\ 3 & 7 \end{vmatrix} + \begin{vmatrix} -2 & 3 \\ -6 & 7 \end{vmatrix} = -4 + 32 + 4 = 32,$$

$$p_3 = (-70 - 18 - 18) - (-6 - 90 - 42) = -106 + 138 = 32.$$

Вычислив соответствующий определитель

$$|A - \lambda E| = \begin{vmatrix} 5-\lambda & -2 & 1 \\ 3 & -2-\lambda & 3 \\ 3 & -6 & 7-\lambda \end{vmatrix} = 0,$$

получаем то же характеристическое уравнение $\lambda^3 - 10\,\lambda^2 + 32\,\lambda - 32 = 0$. Известными нам методами находим его корни (собственные числа, *eigen values*) $\lambda_1 = 2$, $\lambda_2 = \lambda_3 = 4$.

Выбрав $\lambda_1 = 2$, получаем систему уравнений

$$3\,x_1 - 2\,x_2 + x_3 = 0,$$
$$3\,x_1 - 4\,x_2 + 3\,x_3 = 0,$$
$$3\,x_1 - 6\,x_2 + 5\,x_3 = 0,$$

лишь два из которых линейно независимы (ранг равен 2). Следовательно, искомые собственные векторы можно найти не только *с точностью до постоянного множителя*, но и с точностью до константы. Потому задаем одну из переменных равной любой ненулевой константе, например $x_1 = c$, и получаем остальные компоненты собственного вектора $X^{(1)} = [c,\ 3\,c,\ 3\,c\ ]^{\mathrm{T}}$. Для выделения конкретного из найденного семейства векторов обычно либо только фиксируют константу, либо к тому же нормируют, деля его компоненты на длину вектора (в нашем случае при выборе $c = 1$ длина вектора равна на $\sqrt{1^2 + 3^2 + 3^2}$) и получая вектор единичной длины $X^1 = [0.2294\ \ 0.6882\ \ 0.6882]^{\mathrm{T}}$.

При поиске собственных векторов для кратного корня $\lambda_2 = 4$ возникает система тождественных уравнений

$$x_1 - 2\,x_2 + x_3 = 0,$$
$$3\,x_1 - 6\,x_2 + 3\,x_3 = 0,$$
$$3\,x_1 - 6\,x_2 + 3\,x_3 = 0,$$

где решение определяется с точностью до двух констант. Их вы-

бор делается так, чтобы искомые векторы оказались *линейно независимыми*. Например, принимаем $x_2 = c$, $x_3 = c$ и $x_2 = 0$, $x_3 = c$, получая $X^{(2)} = [c\ c\ c]^{\mathrm{T}}$, $X^{(3)} = [c\ 0\ c]^{\mathrm{T}}$ с длиной соответственно $c\sqrt{3}$ и $c\sqrt{2}$. После нормировки кратному собственному значению сопоставляются собственные векторы

$X^{(2)} = [0.5774\ \ 0.5774\ \ 0.5774\ ]^{\mathrm{T}}$ и $X^{(3)} = [-0.7071\ \ 0\ \ -0.7071]^{\mathrm{T}}$.

В общем случае, когда имеются матрица размерности n и корень кратности k, при поиске соответствующих собственных векторов ранг решаемой системы уравнений равен $n - k$ и число возможных независимых решений имеет порядок $C(n, k)$ − числа сочетаний из n по k.

Из приведенного выше примера видим, что не так-то просто найти коэффициенты характеристического уравнения. К тому же поиск всех корней алгебраического уравнения (например, методом Ньютона) является достаточно трудоемкой процедурой. Поэтому стараются искать собственные числа и векторы, минуя построение уравнения.

4.2. Поиск коэффициентов характеристического уравнения

Построение характеристического уравнения в виде алгебраического многочлена (4.3) посредством (4.4), приведенное выше, при больших n требует существенных затрат времени. Примером более удачного подхода к этой задаче может служить изящный *метод Леверрье – Фаддеева* [6, 7], который исключительно просто программируется в различных системах типа MatLab, допускающих работу не только со скалярами, но и массивами. Здесь коэффициенты характеристического уравнения

$$\lambda^n - p_1 \lambda^{n-1} - p_2 \lambda^{n-2} - \ldots - p_{n-1} \lambda - p_n = 0 \qquad (4.5)$$

определяются путем построения последовательности матриц

$$A^{(k)} = A\, B^{(k-1)},\ p_k = \mathrm{Sp}(A^{(k)}) / k,\ B^{(k)} = A^{(k)} - p_k E\ (k = 1, 2, \ldots, n), \qquad (4.6)$$

где $B^{(0)} = E$, $\mathrm{Sp}(A)$ − *след матрицы* (сумма элементов главной диагонали). Обратите внимание на тот факт, что $B^{(n)} = 0$, $B^{(n-1)} / p_n = A^{-1}$ (метод можно использовать и для обращения матрицы).

Отыскав все корни (4.6), *при отсутствии их кратности* ищем матрицу из собственных векторов:

$$X_m = \lambda_m^{n-1} E_m + \lambda_m^{n-2} B_m^{(1)} + \ldots + \lambda_m B_m^{(n-2)} + B_m^{(n-1)},\ m = 1, 2, \ldots, n \quad (4.7)$$

(здесь E_m, $B_m^{(k)}$ – соответствующие столбцы матриц).

Возьмем для примера ранее рассмотренную матрицу и построим последовательность (4.6), дающую в итоге уравнение $\lambda^3 - 10\,\lambda^2 + 32\,\lambda - 32 = 0$:

k	$A^{(k)}$			p_k	$B^{(k)}$		
1	5	−2	1		−5	−2	1
	3	−2	3	**10**	3	−12	3
	3	−6	7		3	−6	−3
2	−28	8	−4		4	8	−4
	−12	0	−12	**−32**	−12	32	−12
	−12	24	−36		−12	24	−4
3	32	0	0				
	0	32	0	**32**			
	0	0	32				

Согласно (4.7), для $\lambda = 2$ получаем векторы с точностью до противоположного знака и нормировки, эквивалентные ранее полученным:

$$X_1 = 2^2 \begin{vmatrix} 1 \\ 0 \\ 0 \end{vmatrix} + 2^1 \begin{vmatrix} -5 \\ 3 \\ 3 \end{vmatrix} + \begin{vmatrix} 4 \\ -12 \\ -12 \end{vmatrix} = \begin{vmatrix} -2 \\ -6 \\ -6 \end{vmatrix},$$

$$X_2 = 2^2 \begin{vmatrix} 0 \\ 1 \\ 0 \end{vmatrix} + 2^1 \begin{vmatrix} -2 \\ -12 \\ -6 \end{vmatrix} + \begin{vmatrix} 8 \\ 32 \\ 24 \end{vmatrix} = \begin{vmatrix} 4 \\ 12 \\ 12 \end{vmatrix},$$

$$X_3 = 2^2 \begin{vmatrix} 0 \\ 0 \\ 1 \end{vmatrix} + 2^1 \begin{vmatrix} 1 \\ 3 \\ -3 \end{vmatrix} + \begin{vmatrix} -4 \\ -12 \\ -4 \end{vmatrix} = \begin{vmatrix} -3 \\ -6 \\ -6 \end{vmatrix}.$$

Заметим, что в обширной литературе недавнего времени, посвященной построению характеристического уравнения, основное внимание уделялось *методам Данилевского**, *А. Н. Крылова* (1863–1945) и ряду других.

* Иван Александрович Лаппо-Данилевский (1896−1931) − советский математик, член-корреспондент АН СССР. Построил теорию функций от матриц и применил ее к проблемам теории линейных дифференциальных уравнений.

Представленный здесь *метод Леверье* − Фаддева***, даже при его негативном отношении к поиску собственных векторов в случае кратных собственных чисел, превосходит вышеназванные методы по простоте компьютерной реализации.

4.3. Степенной метод. Максимальное по модулю собственное значение

В многочисленных итерационных процессах условия сходимости связаны с величиной максимального по модулю собственного значения и соответствующего собственного вектора. Степенной метод [4, 6, 17] предлагает выбрать начальное приближение $X^{(0)}$ для искомого вектора, запустив итерационный процесс

$$Y^{(k)} = A\,X^{(k-1)},\ \rho = \max_{i=1,..,n} Y_i^{(k)},\ X^{(k)} = Y^{(k)} \Big/ \rho,\ k = 1, 2, \ldots \qquad (4.8)$$

(здесь имеется в виду максимальное по абсолютной величине значение). При достаточно большом k величина ρ определяет искомое собственное число, а $X^{(k)}$ − соответствующий собственный вектор.

Если начальное приближение неизвестно, можно взять любой единичный вектор. Однако сходимость к максимальному собственному числу не гарантирована (лучше проверить этот факт путем пересчета с другим начальным приближением, например, ортогональным к ранее выбранному).

Возьмем матрицу

$$A = \begin{vmatrix} 5 & -2 & 1 \\ 3 & -2 & 3 \\ 3 & -6 & 7 \end{vmatrix}.$$

Примем за начальное приближение вектор $X^{(0)} = [1, 0, 0]^{\mathrm{T}}$. Тогда

---

* Урбен Жан Леверье (1811−1877) − французский математик, занимавшийся небесной механикой − вычислением орбиты неизвестной тогда планеты Уран, кометы Лекселя и других небесных тел.

** Дмитрий Константинович Фаддеев (1907−1989) − советский математик, член-корреспондент АН СССР, автор знаменитой монографии, посвященной численным методам линейной алгебры [6].

k	$Y^{(k)}$			ρ	$X^{(k)}$		
1	5	3	3	5	1.0000	0.6000	0.6000
2	4.4000	3.6000	3.6000	4.4000	1.0000	0.8182	0.8182
3	4.1818	3.8182	3.8182	4.1818	1.0000	0.9130	0.9130
		…		…		…	
9	4.0026	3.9974	3.9974	**4.002**6	1.0000	0.9987	0.9987
10	4.0013	3.9987	3.9987	**4.001**3	1.0000	0.9993	0.9993

Нетрудно заметить, что $\rho \to 4$ и $X \to [1, 1, 1]$.

В случае комплексных значений этот метод не работает.

4.4. Метод скалярных произведений. Максимальное по модулю собственное значение симметрической матрицы

Прежде чем приступать к рассмотрению метода скалярных произведений [4, 6, 17], обратим внимание на специфику матрицы собственных векторов для исходной симметрической матрицы. Пусть, например, для матрицы

$$A = \begin{vmatrix} 4 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 5 \end{vmatrix}$$

каким-то способом получены матрица собственных векторов и собственные значения

$$X = \begin{vmatrix} 0.5556 & -0.6691 & \mathbf{0.4936} \\ 0.7729 & 0.1968 & \mathbf{-0.6032} \\ 0.3065 & 0.7166 & \mathbf{0.6265} \end{vmatrix}, \Lambda = \begin{vmatrix} 1.7693 \\ 3.5171 \\ \mathbf{7.7136} \end{vmatrix}.$$

Обратите внимание, что произведение $X^{\mathrm{T}}X = X\,X^{\mathrm{T}} = E$. Отсюда напрашивается вывод, что *собственные векторы образуют ортогональную систему единичных векторов* (ортонормированную систему). Более того, следует обратить внимание на совпадение обратной матрицы с транспонированной $X^{\mathrm{T}} = X^{-1}$.

Метод скалярных произведений, как и степенной, требует задания начального приближения собственного вектора $X^{(0)}$ и следующего итерационного процесса

$$Y^{(k)} = X^{(k)} \Big/ \sqrt{\left(X^{(k)}, X^{k}\right)},\ X^{(k+1)} = A\,Y^{(k)},\ k = 0, 1, 2, \ldots \qquad (4.9)$$

Оценка максимального собственного значения отыскивается как отношение скалярных произведений

$$\lambda = \frac{\left(X^{(k)}, X^{(k)}\right)}{\left(X^{(k)}, X^{(k-1)}\right)}. \qquad (4.10)$$

Обратимся к приведенной выше симметрической матрице и зададим начальное приближение

$$A = \begin{vmatrix} 4 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 5 \end{vmatrix}, \qquad X^{(0)} = \begin{vmatrix} 1 \\ 0 \\ 0 \end{vmatrix}.$$

Тогда получим

k	$X^{(k)}$			λ	$Y^{(k)}$		
1	4	−2	1	5.2500	0.873	−0.436	0.218
2	4.583	−3.928	2.837	7.0226	0.687	−0.589	0.425
3	4.352	−4.581	3.992	7.5624	0.582	−0.613	0.534
4	4.089	−4.684	4.479	7.6820	0.537	−0.611	0.584
5	3.942	−4.681	4.678	7.7070	0.512	−0.608	0.607
6	3.869	−4.669	4.764	7.7122	0.502	−0.606	0.619
7	3.836	−4.661	4.802	7.7133	0.497	−0.604	0.623
8	3.820	−4.657	4.818	7.7135	0.495	−0.604	0.625
9	3.813	−4.655	4.826	7.7136	0.494	−0.604	0.626
10	3.810	−4.654	4.830	**7.7136**	**0.494**	**−0.604**	**0.626**

4.5. Решение проблемы собственных значений в среде MatLab

Средства для вычисления собственных чисел и векторов занимают особое место среди стандартных функций MatLab. Так командами `L=eig(A)` или `[X,L]=eig(A)` получаем вектор L (в виде диагональной матрицы) собственных чисел и матрицу X нормированных собственных векторов-столбцов:

$$A = \begin{vmatrix} 1 & 2 & 3 \\ 1 & 4 & 9 \\ 1 & 8 & 27 \end{vmatrix}, X = \begin{vmatrix} -0.1198 & -0.8484 & -0.7163 \\ -0.3295 & 0.5150 & -0.6563 \\ -0.9365 & -0.1222 & 0.2371 \end{vmatrix},$$

$$L = \begin{vmatrix} 29.9428 & 0 & 0 \\ 0 & 0.2179 & 0 \\ 0 & 0 & 1.8393 \end{vmatrix}.$$

Решение задачи осуществляется на основе QR-алгоритма и его модификаций, но при числе итераций, большем 30 n, может быть прервано с сообщением `Solution will not converge` (решение не сходится).

Для проверки качества поиска (при значительных размерностях и многочисленных особых случаях такая проверка весьма желательна) авторы рекомендуют проверить на близость к нулю значений $A\,X = X\,L$ (для приведенного примера эти значения имеют порядок 10^{-14}).

Собственные значения можно искать как для матрицы A, так и для матричного полинома $A_0 + \lambda\,A_1 + \lambda^2\,A_2 + \ldots + \lambda^p\,A_p$ командой `[R,L]=polyeig(`A_0`,`A_1`,…,`A_p`)`, где `R` – матрица размера $n \times (n \times p)$ собственных векторов.

4.6. Прикладные аспекты

С проблемой собственных значений приходится иметь дело, например, при обработке статистических данных с помощью *метода главных компонент*. Метод используется при разработке тестов в психологии, в геологии и археологии, в экономическом анализе и пр.

Так при обработке статистических данных для выявления их взаимосвязи формируется *симметрическая и положительно определенная матрица R парных коэффициентов корреляции*. Для этой матрицы решением характеристического уравнения $|R - \lambda\,E| = 0$ получаем вектор действительных собственных чисел $\Lambda = \{\lambda_j,\ j = 1, 2, \ldots, n\}$ и *ортонормированные* (взаимно перпендикулярные) собственные векторы (столбцы) $U = \{u_j,\ j = 1, 2, \ldots, n\}$, такие, что выполняется $U^{\mathrm{T}}\,U = U\,U^{\mathrm{T}} = E$.

Например, рассмотрим взаимосвязь между двумя компонентами (случайными последовательностями):

x = [1, 2, 3, 4, 5, 3, 3, …],

y = [10, 18, 26, 34, 42, 24, 28, …]

со средними значениями $\mu_x = 3$, $\mu_y = 26$ и стандартными отклонениями $s_x = 1.2910$, $s_y = 10.3923$. Находим так называемую ковариационную матрицу (см. любой учебник по теории вероятностей)

$$\begin{matrix} 1.6667 & 13.3333 \\ 13.3333 & 108.0000 \end{matrix}$$

и делением соответствующих ее элементов на произведение стандартных отклонений получаем корреляционную матрицу

$$R = \left| \begin{matrix} 1 & 0.9938 \\ 0.9938 & 1 \end{matrix} \right| .$$

Командой `[U,L]=eig(R)` получаем ортонормированные собственные векторы и собственные значения

$$U = \left| \begin{matrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{matrix} \right| , L = \left| \begin{matrix} 0.0062 & 0 \\ 0 & 1.9938 \end{matrix} \right| .$$

Умножая $R_{xy} = 0.9938$ на отношение s_y / s_x, получаем угловой коэффициент $b = 7.9999$ и свободный член $a = \mu_y - b\,\mu_x = 2.0003$, т. е. искомое уравнение регрессии имеет вид $y = 2 + 8\,x$.

Легко видеть (рис. 4.1), что «эллипс рассеивания» сосредоточен вдоль этой линии и существенно сжат вдоль перпендикулярной ей линии, т. е. отклонения от найденной линии регрессии, скорее всего, носят случайный характер.

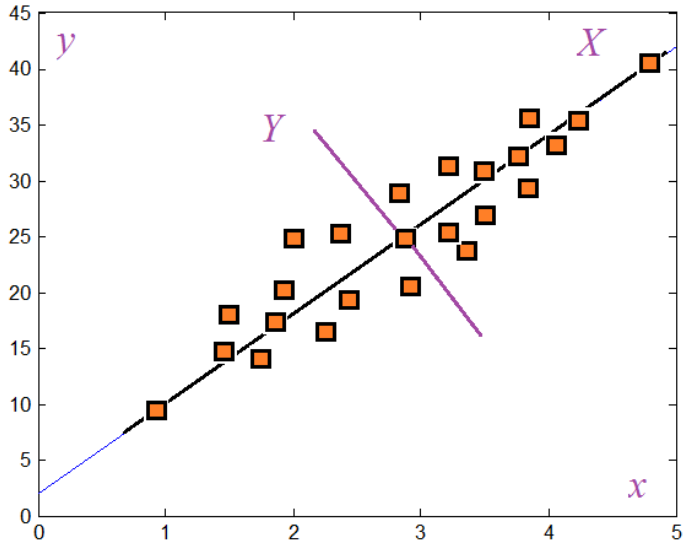


Рис. 4.1

Обратимся к случаю более чем двух компонент. Пусть заданы $m = 4$ случайные последовательности, образующие матрицу

$$x = \begin{vmatrix} 8 & 17 & 18 & 19 & 15 & 13 \\ 5 & 4 & 3 & 2 & 1 & 0 \\ 1 & 4 & 9 & 9 & 4 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{vmatrix},$$

для которых средние значения равны

$$\mu = [15.00, 2.5000, 4.67, 3.50]$$

и стандартные отклонения

$$s = [4.0497, 1.8708, 3.6148, 1.8708].$$

Находим ковариационную матрицу*

$$Cov = \begin{vmatrix} 16.4000 & -2.0000 & 12.4000 & 2.0000 \\ -2.0000 & 3.5000 & 0 & -3.5000 \\ 12.4000 & 0 & 13.0667 & 0 \\ 2.0000 & -3.5000 & 0 & 3.5000 \end{vmatrix}$$

и делением ее строк и столбцов на произведение стандарных отклонений получаем корреляционную матрицу

$$Corr = \begin{vmatrix} 1 & -0.2640 & 0.8471 & 0.2640 \\ -0.2640 & 1 & 0 & -1 \\ 0.8471 & 0 & 1 & 0 \\ 0.2640 & -1 & 0 & 1 \end{vmatrix}.$$

Находим матрицу собственных векторов корреляционной матрицы:

$$U = \begin{vmatrix} 0.0000 & 0.7151 & -0.5088 & 0.4794 \\ 0.7071 & 0.1002 & -0.4003 & -0.5742 \\ -0.0000 & -0.6845 & -0.6486 & 0.3328 \\ 0.7071 & -0.1002 & 0.4003 & 0.5742 \end{vmatrix}$$

и собственные значения $L = [0.0000, 0.1151, 1.6645, 2.2204]$.

Полученные собственные векторы определяют оси в системе координат новых *обобщенных* факторов X, и компоненты вектора U_j определяют долю значимости сочетаний исходных факторов в соответствующей j-й главной компоненте. Например, из вектора U_1 видим, что первая главная компонента определяет-

* В системе MatLab для этой цели имеется процедура `cov(x)`.

ся только значениями второго и четвертого исходных факторов в равных долях.

В новых координатах $X = U^{\mathrm{T}} x$ исходные последовательности приводятся к виду

X_1 = [4.2426, 4.2426, 4.2426, 4.2426, 4.2426, 4.2426],

X_2 = [5.4371, 9.6191, 6.7113, 7.2260, 7.5877, 8.0106],

X_3 = [–6.3202, –12.0446, –14.9958, –14.7040, –8.6252, –4.8612],

X_4 = [1.8712, 8.3326, 11.6244, 13.2522, 10.8190, 10.0102].

Собственные значения λ_j выступают как оценки разброса (дисперсии) вдоль соответствующих полуосей преобразованного «эллипса рассеивания», и отношение $\lambda_j / \sum_{j=1}^{m} \lambda_j$ определяет вклад j-й *главной компоненты* в общую дисперсию исходных факторов. Упорядочив собственные числа в порядке уменьшения λ_j и установив «порог» (на уровне 5–10 %), обнаруживаем приемлемое количество значимых главных компонент $q < m$.

В нашем случае очевидна ничтожность влияния первых двух главных компонент, и без ущерба от них можно отказаться. Обнулив первые два столбца в матрице X^{T} и вычислив произведение U на X^{T}, возвращаемся к модифицированной исходной матрице

$$x_{\text{корр}} = \begin{vmatrix} 1.3231 & 5.8920 & 8.2196 & 9.3706 & 7.6501 & 7.0782 \\ 4.1387 & 7.4096 & 9.0999 & 8.7370 & 4.8199 & 2.3245 \\ 4.8483 & 11.1477 & 14.3795 & 14.8419 & 9.9252 & 7.1601 \\ -1.0289 & 0.7761 & 1.6841 & 2.7159 & 3.3418 & 4.1300 \end{vmatrix},$$

которую и подвергают дальнейшим исследованиям.

Другим примером необходимости поиска собственных значений может служить известная модель Леонтьева, сводящаяся к решению системы $X = A\,X + Y$, где X – вектор всей производимой продукции, A − матрица норм расхода на воспроизводство, Y − товарная продукция. Легко увидеть $X = (E - A)^{-1} Y$. Как искать обратную матрицу, если ее размерность определяется сотнями или тысячами? Иногда можно заменить $(E - A)^{-1} = E + A + A^2 + \ldots + A^m + \ldots$, но для сходимости ряда тре-

буется уверенность в том, что норма матрицы $\|A\|_2 = \sqrt{\lambda_{\max}(AA^{\mathrm{T}})} < 1.$

Не вдаваясь в ее происхождение, при решении системы однородных дифференциальных уравнений $\frac{dy_i}{dx} = \sum_{j=1}^{n} a_{ij} y_j, i = \overline{1, n}$ (в матричной записи $\frac{dY}{dx} = AY$) берут $y_i = z_i \, e^{\lambda x}$ и с учетом $e^{\lambda x} \neq 0$ получают однородную систему линейных алгебраических уравнений $A\,Z = \lambda\,Z$, ненулевые решения которой и представляют интерес.

Самым популярным методом поиска всех собственных значений и собственных векторов симметрической матрицы (для таких матриц собственные значения всегда вещественны) является *метод вращений* (*Якоби*) с различными модификациями, описание которого можно найти в многочисленных публикациях.

Существуют и другие эффективные подходы к решению проблемы собственных значений. Среди них популярен *метод Гивенса* приведения симметрической матрицы к трехдиагональной, использующий ту же идею вращений.

Глава 5. АППРОКСИМАЦИЯ ДАННЫХ

Задача аппроксимации данных возникает в самых разнообразных случаях. Камнедробилка при работе в течение нескольких последовательных интервалов времени дает постепенно уменьшаемый выход товарной продукции (гравия). Требуется оценить длительность сеанса ее работы, при которой стоимость произведенной продукции не превысит затрат на электроэнергию и обслуживание (рис. 5.1). Для простоты и точности анализа хотелось бы найти аналитическое выражение зависимости выхода продукции от длительности сеанса. Группа из 100 студентов измерила длину бассейна с точностью до полуметра (многое зависит от инструмента, аккуратности измерителя и пр.), сравнила ее с плановой. Случайны ли эти ошибки (рис. 5.2) или допущены нарушения при строительстве?

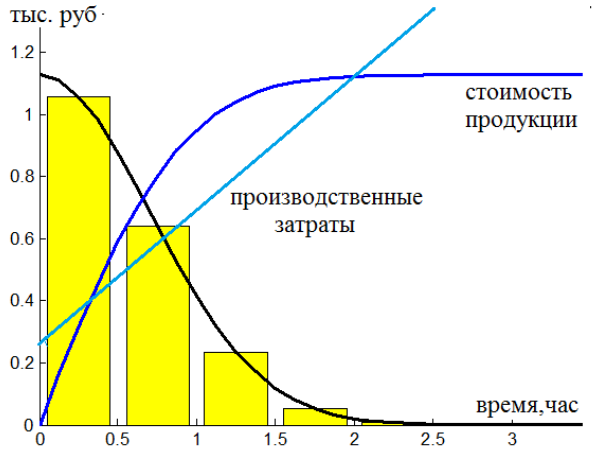


Рис. 5.1

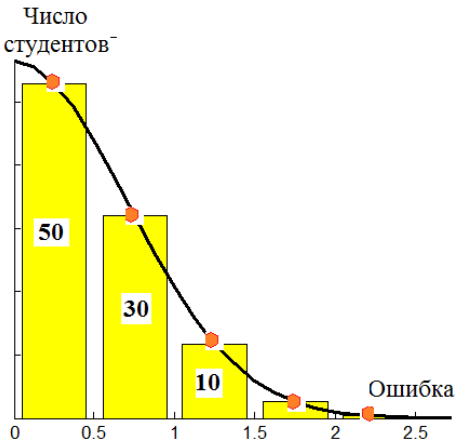


Рис. 5.2

Вычисление значений функций со сложным аналитическим описанием или заданием в табличной форме является одной из распространенных задач прикладной математики. Например, специалисты в области математической физики при решении задачи теплообмена в трубе пришли к необходимости вычисления значений так называемых бесселевых функций. Известные для них представления степенными рядами часто сходятся медленно, и потому желательно обладать более простыми, приемлемыми как по затратам времени, так и по точности значениями (разумеется, подобные представления уже получены в результате значительных усилий заинтересованных математиков-прикладников).

Часто мы сталкиваемся с табличным заданием функций. Даже школьники вынуждены искать значения синусов или логарифмов в промежутках между узлами таблиц. Хорошо, если значения функции в таблице получены точно и прошли проверку временем. А если они представляют результаты какого-то эксперимента в условиях случайных воздействий? Хорошо, если они не связаны с так называемым человеческим фактором.

Вообще говоря, *задача точного восстановления функции по ее табличным значениям некорректна.* Если потребовать, чтобы значения этой функции совпадали с табличными значениями, то можно подобрать множество таких функций (рис. 5.3). Поэтому перед вычислителем, решающим задачу восстановления функции, возникают проблемы, связанные с выбором *класса аппроксимирующих функций*, *точности аппроксимации* и *критерия согласия* между функцией и исходными данными.

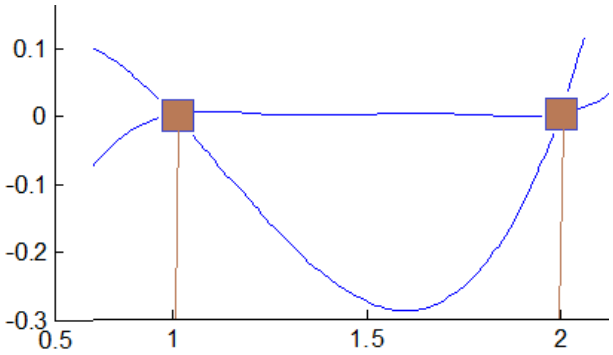


Рис. 5.3

Обычно на практике используют классы самых простейших функций:

1) линейные комбинации функций 1, x, x^2, …, x^n, т. е. функции из класса полиномов степени не выше *числа узлов* (аппроксимация алгебраическим многочленом заданной степени);

2) линейные комбинации функций $\sin(a_k x)$ и $\cos(a_k x)$ (аппроксимация тригонометрическим многочленом, или отрезком ряда Фурье);

3) комбинации экспоненциальных функций $\exp(\lambda_k x)$ с вышеуказанными и некоторые другие.

В качестве критерия согласия используют три условия:

1) точное совпадение значений искомой функции с «экспериментом», со значениями в узлах таблицы (*критерий интерполяции*);

2) сумма квадратов отклонений значений искомой и табличной функций минимальна (*критерий среднеквадратической аппроксимации*);

3) максимальное по абсолютной величине отклонение значений искомой и табличной функций минимально (*критерий равномерной аппроксимации*).

5.1. Среднеквадратическая аппроксимация

5.1.1. Метод наименьших квадратов

Пусть имеется таблица, содержащая N значений аргумента x_i и соответствующих им значений функции F_i (эта таблица могла возникнуть при вычислениях некоторой аналитически заданной функции на некоторой равномерной или неравномерной сетке значений аргумента, при проведении эксперимента по выявлению зависимости силы тока от сопротивления в электрической сети, при выявлении связи между солнечной активностью и количеством обращений в кардиологический центр, между размером дотаций в сельское хозяйство и объемом производства сельхозпродукции и т. п.). Поставим задачу поиска функции из класса алгебраических многочленов m-го порядка

$$F(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m \qquad (5.1)$$

такой, что сумма квадратов отклонений ее от табличной функции

$$R(a_0, a_1, a_2, ..., a_m) = \sum_{i=1}^{N} \left[F(x_i) - F_i\right]^2 \qquad (5.2)$$

является минимальной.

Очевидно, что частные производные функции R при оптимальном выборе неизвестных коэффициентов многочлена должны обращаться в нуль, откуда возникает система уравнений

$$\frac{\partial R}{\partial a_k} \equiv 2 \sum_{i=1}^{N} \left[F(x_i) - F_i\right] \frac{\partial F}{\partial a_k} = 0, \; k = 0, 1, 2, \ldots, m \qquad (5.3)$$

или

$$\frac{\partial R}{\partial a_k} \equiv 2 \sum_{i=1}^{N} \left[a_0 + a_1 x_i + a_2 x_i^2 + ... + a_m x_i^m - F_i\right] x_i^k = 0, k = 0, 1, 2, \ldots, m.$$

Если обозначить через

$$\frac{1}{N} \sum_{i=1}^{N} x_i^k = \overline{x^k}, \qquad (5.4)$$

то из (5.3) с учетом (5.1) возникает система линейных алгебраических уравнений с симметрической матрицей коэффициентов, определитель которой отличен от нуля (если значения x_i не равны константе), что гарантирует существование и единственность ее решения (для решения можно воспользоваться любым из известных методов, в частности, методом квадратных корней):

$$\begin{aligned}
&a_0 + a_1\overline{x} + a_2\overline{x^2} + \ldots + a_m\overline{x^m} = \overline{F},\\
&a_0\overline{x} + a_1\overline{x^2} + a_2\overline{x^3} + \ldots + a_m\overline{x^{m+1}} = \overline{F\,x},\\
&a_0\overline{x^2} + a_1\overline{x^3} + a_2\overline{x^4} + \ldots + a_m\overline{x^{m+2}} = \overline{F\,x^2},\\
&\qquad\qquad \ldots\ldots\ldots\ldots\\
&a_0\overline{x^m} + a_1\overline{x^{m+1}} + a_2\overline{x^{m+2}} + \ldots + a_m\overline{x^{2m}} = \overline{F\,x^m}.
\end{aligned} \tag{5.5}$$

При аппроксимации линейной функцией $F(x) = a + b\,x$ имеем

$$b = \frac{\overline{F\,x} - \overline{F}\,\overline{x}}{\overline{x^2} - \overline{x}^2},\ a = \overline{F} - b\,\overline{x}. \tag{5.6}$$

При выборе квадратичной функции $F(x) = a + b\,x + c\,x^2$ коэффициенты получаются из системы

$$\begin{aligned}
&a + b\,\overline{x} + c\,\overline{x^2} = \overline{F},\\
&a\,\overline{x} + b\,\overline{x^2} + c\,\overline{x^3} = \overline{F\,x},\\
&a\overline{x^2} + b\,\overline{x^3} + c\,\overline{x^4} = \overline{F\,x^2}.
\end{aligned} \tag{5.7}$$

Пример. Пусть задана табличная функция

x	0	0.1	0.2	0.3	0.4	0.5	0.6
$f(x)$	2.0000	1.9997	1.9977	1.9928	1.9841	1.9712	1.9538

x	0.7	0.8	0.9	1.0
$f(x)$	1.9317	1.9052	1.8743	1.8394

Попытка подбора аппроксимирующего многочлена соответствующей степени k приводит к таблице коэффициентов (коэффициенты приведены по убыванию показателей степеней)

k						
0	1.9500					
1	–0.1577	2.0288				
2	–0.2061	0.0484	1.9979			
3	–0.0387	–0.1480	0.0263	1.9993		
4	0.0870	–0.2128	–0.0392	0.0045	1.9999	
5	–0.0430	0.1945	–0.3066	–0.0059	0.0005	2.0000

Отобразив полученные аппроксимации (рис. 5.4), видим, что графики полиномов 2- и 3-го порядков практически совпадают. Это можно установить, не прибегая к визуализации, если оценить так называемую *остаточную дисперсию* (среднеквадратическую характеристику разброса данных относительно получаемой аппроксимации) $D_0 = 0.0028$, $D_1 = 3.3257 \cdot 10^{-04}$, $D_2 = 3.3144 \cdot 10^{-04}$, $D_3 = 8.4229 \cdot 10^{-07}$. Если начальная дисперсия равна 0.0028, то уже аппроксимация полиномом первого порядка уменьшает дисперсию в 9 раз, а второго порядка сопоставима с первой.

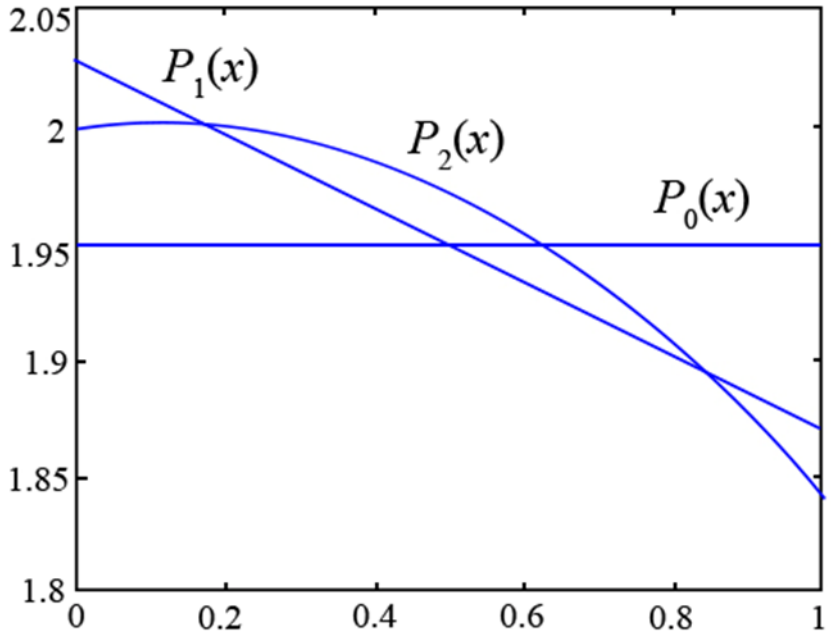


Рис. 5.4

Аналогично можно получить и аппроксимации, отличные от полиномиальной. Так, для линейно-логарифмической функции

$$F(x) = a + b \ln(x) \qquad (5.8)$$

поиск коэффициентов сведется к решению системы двух линейных уравнений, откуда получаем

$$b = \frac{\overline{F\,\ln(x)} - \overline{F}\;\overline{\ln(x)}}{\overline{\ln^2(x)} - \overline{\ln(x)}^2},\; a = \overline{F} - b\,\overline{\ln(x)}. \qquad (5.8')$$

Если взять функцию, *нелинейную* относительно коэффициентов, например $F(x) = a\,x^b$, то система (5.3) сведется к нелинейной относительно b системе

$$\begin{aligned} a\sum_{i=1}^{N} x_i^{2b} &= \sum_{i=1}^{N} F_i\; x_i^b, \\ a\sum_{i=1}^{N} x_i^{2b}\ln(x_i) &= \sum_{i=1}^{N} F_i\; x_i^b\;\ln(x_i). \end{aligned} \qquad (5.9)$$

Можно выполнить искусственную «линеаризацию» заменой $\ln(F(x)) = \ln(a) + b\,\ln(x)$. Тогда можно воспользоваться стандартным подходом, дающим

$$b = \frac{\overline{\ln(F)\ln(x)} - \overline{\ln(F)}\;\overline{\ln(x)}}{\overline{\ln^2(x)} - \overline{\ln(x)}^2},\;\; a = \exp\left(\overline{\ln(F)} - b\,\overline{\ln(x)}\right). \qquad (5.9')$$

Но, например, для функции $F(x) = a\,x^b + c$ избавление от нелинейности нереально.

Совершенно аналогичный подход можно использовать и для аппроксимации функций нескольких переменных. Так линейная аппроксимация

$$F(x_1, x_2, \ldots, x_n) = a_0 + a_1\,x_1 + a_2\,x_2 + \ldots + a_n\,x_n \qquad (5.10)$$

приводит к линейной системе

$$\begin{aligned} &a_0 + a_1\overline{x_1} + a_2\overline{x_2} + \ldots + a_n\overline{x_n} = \overline{F}, \\ &a_0\overline{x_1} + a_1\overline{x_1^2} + a_2\overline{x_1x_2} + \ldots + a_n\overline{x_1x_n} = \overline{F\;x_1}, \\ &a_0\overline{x_2} + a_1\overline{x_2x_1} + a_2\overline{x_2^2} + \ldots + a_n\overline{x_2x_n} = \overline{F\;x_2}, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_0\overline{x_n} + a_1\overline{x_nx_1} + a_2\overline{x_nx_2} + \ldots + a_n\overline{x_n^2} = \overline{F\;x_n}. \end{aligned} \qquad (5.11)$$

Часто используется и так называемая *мультипликативная* аппроксимация

$$F(x_1, x_2, \ldots, x_n) = a_0\;x_1^{a_1}x_2^{a_2}\ldots x_n^{a_n}, \qquad (5.12)$$

поиск коэффициентов которой предварительным логарифмированием сведется к решению системы линейных уравнений.

Использованный выше *метод наименьших квадратов* служит основой методов статистической обработки результатов наблюдений. Величины $\frac{1}{N}\sum_{i=1}^{N}\left[F(x_i)-F_i\right]^2$ и $\frac{1}{N}\sum_{i=1}^{N}\left(F_i-\overline{F}\right)^2$ представляют собой остаточную и исходную дисперсии – оценки качества полученной аппроксимации.

5.1.2. Среднеквадратическая аппроксимация на интервале

Рассматриваемая задача связана с аппроксимацией аналитически заданной функции другой, более простой и эффективной по вычислительным затратам функцией. Такая аппроксимация используется при создании библиотек популярных функций или при необходимости массовых вычислений. Если вы встречаетесь с сообщением о существовании возможности замены некоторой функции конкретными алгебраическими или тригонометрическими многочленами в каком-то диапазоне значений аргумента с указанной точностью, то наверняка за этой аппроксимацией стоят приведенные ниже методы.

5.1.2.1. Аппроксимация алгебраическими многочленами

Заменим некоторую функцию $F(x)$, заданную на отрезке $[a, b]$, например, алгебраическим многочленом

$$P_m(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m, \qquad (5.13)$$

потребовав минимума его отклонения от исходной функции в смысле среднеквадратического приближения

$$R(a_0, a_1, a_2, \ldots, a_m) = \int_a^b \left[P_m(x) - F(x)\right]^2 dx \qquad (5.14)$$

среди всех алгебраических многочленов степени не выше m.

Из равенства нулю частных производных R по a_i получаем

$$\sum_{i=0}^{m} a_i \int_a^b x^{i+k} dx \equiv \sum_{i=0}^{m} a_i \frac{b^{i+k+1} - a^{i+k+1}}{i+k+1} = \int_a^b F(x)\, x^k dx\,, k = \overline{0,m}. \qquad (5.15)$$

Например, если взять $a = -1$, $b = 1$, то при $m = 4$ имеем систему

$$a_0 + \tfrac{1}{3}a_2 + \tfrac{1}{5}a_4 = \tfrac{1}{2}\int_{-1}^{1} F(x)\,dx = I_0,$$

$$\tfrac{1}{3}a_1 + \tfrac{1}{5}a_3 = \tfrac{1}{2}\int_{-1}^{1} x\cdot F(x)\,dx = I_1,$$

$$\tfrac{1}{3}a_0 + \tfrac{1}{5}a_2 + \tfrac{1}{7}a_4 = \tfrac{1}{2}\int_{-1}^{1} x^2 F(x)\,dx = I_2,$$

$$\tfrac{1}{5}a_1 + \tfrac{1}{7}a_3 = \tfrac{1}{2}\int_{-1}^{1} x^3 F(x)\,dx = I_3,$$

$$\tfrac{1}{5}a_0 + \tfrac{1}{7}a_2 + \tfrac{1}{9}a_4 = \tfrac{1}{2}\int_{-1}^{1} x^4 F(x)\,dx = I_4,$$

решение которой

$$a_0 = \tfrac{15}{64}(15\,I_0 - 70\,I_2 + 63 I_4),\ a_1 = \tfrac{15}{4}(5\,I_1 - 7\,I_3),$$
$$a_2 = \tfrac{105}{32}(-5\,I_0 + 42\,I_2 - 45\,I_4), a_3 = \tfrac{35}{4}(-3\,I_1 + 5\,I_3), \qquad (5.16)$$
$$a_4 = \tfrac{315}{64}(3\,I_0 - 30\,I_2 + 35\,I_4)$$

(однократное вычисление приведенных выше интегралов, как мы увидим далее, не составит затруднений).

Если взять теперь конкретную функцию, например $F(x) = x^{1/3}$, и вычислить значения $I_0 = I_2 = I_4 = 0$, $I_1 = 3/7$, $I_3 = 3/13$, то можно найти коэффициенты полинома и получить аппроксимацию данной функции на отрезке $[-1, 1]$ (рис. 5.5) $x^{1/3} \cong 1.978\,x - 1.1538\,x^3$ (найдена весьма грубая аппроксимация).

5.1.2.2. Аппроксимация ортогональными многочленами

Рассмотрим задачу аппроксимации обобщенным многочленом

$$P_m(x) = \sum_{i=0}^{m} a_i \varphi_i(x), \qquad (5.17)$$

где $\varphi_i(x)$ − система линейно независимых и ортогональных на $[a, b]$ с весом $\rho(x)$ функций

$$(\varphi_i, \varphi_k) = \int_a^b \rho(x)\,\varphi_i(x)\,\varphi_k(x)\,dx = \begin{cases} 0, & i \neq k \\ T \neq 0, & i = k \end{cases}. \qquad (5.18)$$

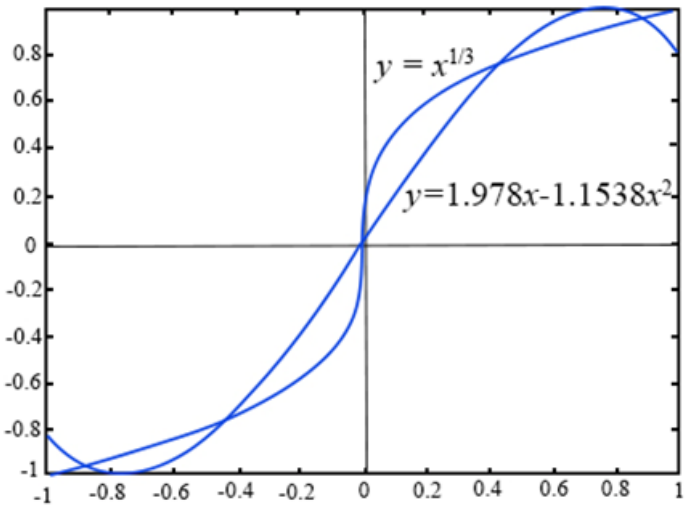


Рис. 5.5

Минимизация

$$R(a_0, a_1, a_2, \ldots, a_m) = \int_a^b \rho(x)\left[P_m(x) - F(x)\right]^2 dx \qquad (5.19)$$

на классе обобщенных многочленов приводит к системе

$$\sum_{i=0}^{m} a_i(\varphi_i(x), \varphi_k(x)) = (F(x), \varphi_k(x)),\ k = \overline{0, m}.$$

Отсюда

$$a_k = \frac{(F, \varphi_k)}{(\varphi_k, \varphi_k)} = \frac{\int_a^b \rho(x)\,F(x)\,\varphi_k(x)\,dx}{\int_a^b \rho(x)\,\varphi_k^2(x)\,dx}. \qquad (5.20)$$

Рассмотрим ряд известных в приложениях систем функций. *Система тригонометрических функций* 1, cos(x), sin(x), cos(2 x), sin(2 x), …, cos(m x), sin(m x) является ортогональной на отрезке [0, 2 π]:

$$\int_0^{2\pi} \sin(kx)\cos(nx)\,dx = 0,$$

$$\int_0^{2\pi} \sin(kx)\sin(nx)\,dx = \begin{cases} 0, & k \neq n, k = n = 0 \\ \pi, & k = n \neq 0 \end{cases},$$

$$\int_0^{2\pi} \cos(kx)\cos(nx)\,dx = \begin{cases} 0, & k \neq n \\ \pi, & k = n \neq 0 \\ 2\pi, & k = n = 0 \end{cases}.$$

С учетом (5.20) при $\rho(x) = 1$ получаем аппроксимацию *тригонометрическим многочленом*, или *отрезком ряда Фурье*:

$$P_m(x) = a_0 + \sum_{k=1}^{m} [a_k \cos(kx) + b_k \sin(kx)], \qquad (5.21)$$

где

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} F(x)\,dx,\ a_k = \frac{1}{\pi}\int_0^{2\pi} F(x)\cos(kx)dx,\ k = \overline{1,m},$$
$$b_k = \frac{1}{\pi}\int_0^{2\pi} F(x)\sin(kx)dx,\ k = \overline{1,m}. \qquad (5.22)$$

Как правило, аппроксимация Фурье эффективна при исследовании колебательных процессов не только в технике, но и в разнообразных экономических и социологических исследованиях.

Общеизвестны ортогональные многочлены − полиномы Чебышева*, Лежандра, Эрмита, Лягерра и т. д. [2].

Полиномы Чебышева первого рода определяются в виде

$$T_n(x) = \cos(n \arccos(x)), \qquad (5.23)$$

откуда $T_0(x) = 1$, $T_1(x) = x$, $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$, $T_4(x) = 8x^4 - 8x^2 + 1$, $T_5(x) = 16x^5 - 20x^3 + 5x$, ... Можно воспользоваться удобным рекуррентным соотношением

$$T_0(x) = 1,\ T_1(x) = x,\ T_n(x) = 2x\,T_{n-1}(x) - T_{n-2}(x). \qquad (5.23')$$

Эти полиномы ортогональны на [–1, 1] с весом $1/\sqrt{1-x^2}$

---

* Пафнутий Львович Чебышев (1821−1894) − русский математик и механик, один из основоположников теории приближения функций. Известен работами в теории чисел, теории вероятностей, численном интегрировании, механике.

$$\int_{-1}^{1} \frac{T_n(x)\,T_k(x)}{\sqrt{1-x^2}}dx = \begin{cases} 0, & k \neq n \\ \pi\ , & k = n = 0\,, \\ \frac{\pi}{2}\ , & k = n \neq 0 \end{cases} \qquad (5.24)$$

откуда для аппроксимации

$$P_m(x) = \sum_{k=0}^{m} a_k\, T_k(x) \qquad (5.25)$$

имеем

$$a_0 = \frac{1}{\pi}\int_{-1}^{1} \frac{F(x)}{\sqrt{1-x^2}}\,dx,\ a_k = \frac{1}{2\pi}\int_{-1}^{1} \frac{F(x)\,T_k(x)}{\sqrt{1-x^2}}dx\ ,\ k = \overline{1,m}. \qquad (5.25')$$

При $m = 4$ и $F(x) = |x|$ после нахождения коэффициентов $a_0 = 2/\pi$, $a_2 = -1/3\ \pi$, $a_4 = -4/15\ \pi$, $a_1 = a_3 = 0$ получаем

$$P_4(x) = \frac{2}{15\pi}\ (-16x^4 + 30x^2 + 3)\,.$$

Заметим, что приближение по Чебышеву, благодаря своей весовой функции, обеспечивает бóльшую точность в середине интервала, тогда как при $\rho(x) = 1$ все точки интервала равноценны (рис. 5.6).

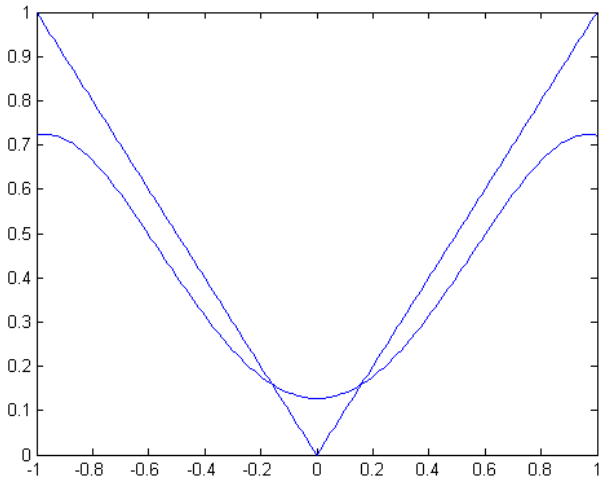


Рис. 5.6

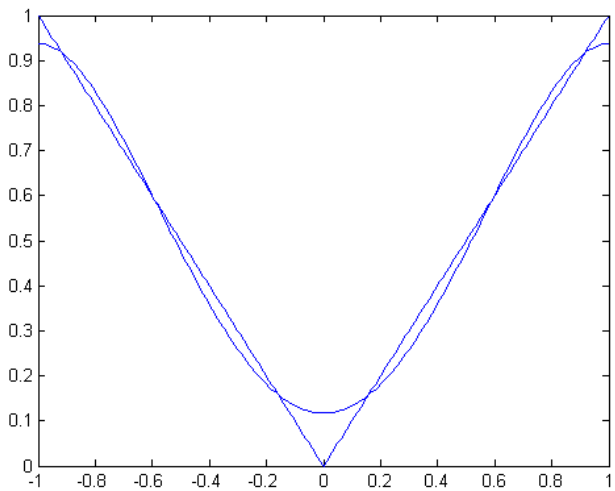


Рис. 5.7

Полиномы Чебышева второго рода

$$U_n(x) = \frac{\sin\big((n+1)\arccos(x)\big)}{\sqrt{1-x^2}} \qquad (5.26)$$

дают аппроксимацию $P_m(x) = \sum_{k=0}^{m} a_k U_k(x)$, где

$$a_k = \frac{2}{\pi}\int_{-1}^{1} F(x)U_k(x)\sqrt{1-x^2}\,dx = \frac{1}{\pi}\int_{-1}^{1} F(\cos(t))\sin(t)\sin((k+1)t)\,dt$$

обеспечивают повышенную точность вблизи концов интервала.

Полиномы Лежандра[*] (частный случай так называемых *сферических функций*) определяются в форме

$$L_n(x) = \frac{1}{2^n\, n!}\frac{d[(x^2-1)^n]}{dx^n}. \qquad (5.27)$$

Откуда находим

$$L_0(x)=1,\ L_1(x)=x,\ L_2(x)=\frac{3x^2-1}{2},\ L_3(x)=\frac{5x^3-3x}{2},$$

$$L_4(x)=\frac{35x^4-30x^2+3}{8},\ L_5(x)=\frac{63x^5-70x^3+15x}{8},\ldots$$

Легко установить рекуррентное соотношение

$$L_{n+1}(x) = \tfrac{2n+1}{n+1}\,x\,L_n(x) - \tfrac{n}{n+1}\,L_{n-1}(x),\ n\ge 1. \qquad (5.27')$$

Можно показать, что

$$\int_{-1}^{1} L_k(x)\,L_n(x)\,dx = \begin{cases} 0, & k\ne n \\ \frac{2}{2k+1}, & k=n \end{cases}. \qquad (5.27'')$$

Откуда для аппроксимации $P_m(x) = \sum_{k=0}^{m} a_k\,L_k(x)$ имеем

$$a_k = \tfrac{2k+1}{2}\int_{-1}^{1} F(x)\,L_k(x)\,dx,\ k=\overline{0,m}. \qquad (5.27''')$$

Так при $m = 4$ и $F(x) = |x|$ получаем (рис. 5.7)

$$|x| \cong P_4(x) = \tfrac{1}{2} + \tfrac{5}{8}\,\tfrac{3x^2-1}{2} - \tfrac{3}{16}\,\tfrac{35x^4-30x^2+3}{8} = \tfrac{15}{128}\,(-7x^4+14x^2+1)$$

---

* Адриен Мари Лежандр (1752−1833) − французский математик, вместе с Лагранжем и Лапласом участвовал в измерении длины одного градуса меридиана между Дюнкерком и Барселоной через Париж для создания эталона метра (сорокамиллионная часть длины этого отрезка меридиана). После проведения измерения длины меридиана был изготовлен эталон метра. Развил теорию геодезических измерений, сферическую тригонометрию. В области математического анализа введены многочлены и преобразование Лежандра, исследованы эйлеровы интегралы. Автор метода наименьших квадратов (Гаусс открыл этот метод независимо и раньше Лежандра, но опубликовал позднее).

(полученный многочлен 4-й степени в смысле среднеквадратического отклонения ближе к исходной функции, чем непосредственная аппроксимация многочленом той же степени).

Для аппроксимации при $-\infty < x < \infty$ можно воспользоваться *полиномами Эрмита*[*] (*функциями параболического цилиндра*)

$$H_n(x) = (-1)^n \mathrm{e}^{x^2} \frac{d}{dx^n}\left(\mathrm{e}^{-x^2}\right), \qquad (5.28)$$

первые из которых равны $H_0(x) = 1$, $H_1(x) = 2x$, $H_2(x) = 4x^2 - 2$, $H_3(x) = 8x^3 - 12x$, $H_4(x) = 16x^4 - 48x^2 + 12$, $H_5(x) = 32x^5 - 160x^3 + 120x$, …, связанных рекуррентным соотношением

$$H_{n+1}(x) = 2x\,H_n(x) + 2n\,L_{n-1}(x),\ n \geq 1,$$

ортогональных на $(-\infty, \infty)$ с весом e^{-x^2}

$$\int_{-\infty}^{\infty} \mathrm{e}^{-x^2} H_m(x)\,H_n(x)\,dx = \begin{cases} 2^n\,n!\sqrt{\pi}, & m = n \\ 0, & m \neq n \end{cases} \qquad (5.28')$$

и при аппроксимации $P_m(x) = \sum_{k=0}^{m} a_k\,H_k(x)$ дающих

$$a_k = \frac{1}{2k\,k!\sqrt{\pi}} \int_{-\infty}^{\infty} \mathrm{e}^{-x^2} F(x)\,H_k(x)\,dx,\ k = \overline{0,m}. \qquad (5.28'')$$

Аппроксимация полиномами Эрмита наиболее точна в окрестности нуля (здесь весовая функция минимальна).

При аппроксимации на интервале $[0, \infty)$ прибегают к *полиномам Лагерра*[**]

$$L_n(x) = (-1)^n \mathrm{e}^{x} \frac{d}{dx^n}\left(x^n \mathrm{e}^{-x}\right), \qquad (5.29)$$

где $L_0(x) = 1$, $L_1(x) = x - 1$, $L_{n+1}(x) = (x - 2n - 1)\,L_n(x) - n^2\,L_{n-1}(x)$, $n \geq 1$, ортогональных с весом e^{-x}

---

* Шарль Эрмит (1822−1901) − французский математик, признанный лидер математиков Франции во второй половине XIX века. Основные работы в теории чисел, квадратичных форм, теории инвариантов, ортогональных многочленов, эллиптических функций и алгебре.

** Эдмон Никола Лагерр (1834−1886) − французский математик. Труды по геометрии, комплексному анализу. Исследовал ортогональные многочлены.

$$\int_0^{\infty} e^{-x} L_m(x)\, L_n(x)\, dx = \begin{cases} (n!)^2, & m = n \\ 0\ , & m \neq n \end{cases}$$

и при аппроксимации $P_m(x) = \sum_{k=0}^{m} a_k\ L_k(x)$ дающих

$$a_k = \frac{1}{(k!)^2} \int_0^{\infty} e^{-x}\ F(x)\, L_k(x)\, dx,\ \ k = \overline{0, m}. \qquad (5.29')$$

5.1.2.3. Аппроксимация табличных функций на интервале

Ранее мы рассмотрели детали аппроксимации таблично заданной функции алгебраическим многочленом определенной степени. Некоторые преимущества дает использование ортогональных многочленов.

Пусть задана таблица N значений функции $F(x)$ для аргумента, изменяющегося в диапазоне от 0 до 2 π. Аппроксимируем ее тригонометрическим многочленом m-й степени (необходимо выполнение условия $2m + 1 < N$):

$$P_m(x) = a_0 + \sum_{k=1}^{m} \left[a_k \cos(kx) + b_k \sin(kx) \right].$$

Поставив задачу среднеквадратической аппроксимации и выполнив необходимые преобразования, получаем так называемые *формулы Бесселя*[*]

$$a_0 = \frac{1}{N} \sum_{i=1}^{N} F(x_i),\ a_k = \frac{2}{N} \sum_{i=1}^{N} F(x_i) \cos(kx_i),$$

$$b_k = \frac{2}{N} \sum_{i=1}^{N} F(x_i) \sin(kx_i),\ \ k = \overline{1, m}.$$

Если обратиться к чебышевской аппроксимации (5.25) для функции, таблично заданной в отрезке [–1, 1], то ее коэффициенты

---

[*] Фридрих Вильгельм Бессель (1784−1846) − немецкий математик и астроном. При решении задач астрономии развил некоторые методы численного анализа. Особую известность приобрел в решении задач математической физики, в частности задачи теплопроводности в цилиндрических координатах.

$$a_k = \frac{\sum\limits_{i=1}^{N} F(x_i)\, T_k(x_i)}{\sum\limits_{i=1}^{N} [T_k(x_i)]^2},\ k = \overline{0, m}.$$

5.1.2.4. Сглаживание табличных функций

Пусть имеется таблица значений функции $F(x)$ для равноотстоящих значений аргумента $x_k = x_0 + k\,h$, $k = 0, 1, 2, \ldots, N$. Может быть поставлена задача приближения всех или части этих значений полиномом заданной степени m, т. е. замены $F(x_i)$ значением $f(x_i)$ на основе известных n значений F слева и справа ($m \le 2\,n$) без поиска самого полинома.

Примерами такого приближения (сглаживания) для внутренних узлов таблицы могут служить

$m = 1$, $n = 1$: $f(x_i) = \{F(x_{i-1}) + F(x_i) + F(x_{i+1})\} / 3$;

$m = 1$, $n = 3$: $f(x_i) = \{F(x_{i-2}) + F(x_{i-1}) + F(x_i) + F(x_{i+1}) + F(x_{i+2})\} / 5$;

$m = 3$, $n = 2$: $f(x_i) = \{-3\,F(x_{i-2}) + 12\,F(x_{i-1}) + 17\,F(x_i) + 12\,F(x_{i+1}) - 3\,F(x_{i+2})\} / 35$;

$m = 5$, $n = 3$: $f(x_i) = \{5\,F(x_{i-3}) - 30\,F(x_{i-2}) + 75\,F(x_{i-1}) + 131\,F(x_i) + 75\,F(x_{i+1}) - 30\,F(x_{i+2}) + 5\,F(x_{i+3})\} / 231$.

Можно проводить сглаживание несколько раз, но многократное сглаживание способно исказить представление исходной функции.

При прогнозировании какой-то величины на основе временного ряда часто используется так называемое *простое экспоненциальное сглаживание*

$$F_t = \alpha\, Y_t + (1 - \alpha)\, F_{t-1},$$

где F_t − сглаженный ряд; Y_t − исходный ряд; α − коэффициент сглаживания, выбираемый априори с учетом точности значений исходного ряда и внешних условий ($0 < \alpha < 1$).

5.2. Равномерная аппроксимация

Задача равномерной аппроксимации существенно сложнее среднеквадратической. Выше мы ставили и успешно решали за-

дачу поиска многочлена заданной степени, имея возможность оценить качество полученной аппроксимации. Здесь, даже заменив исходную функцию $F(x)$ на $[a, b]$ отрезком ряда Тейлора − многочленом $P_m(x)$ при соблюдении условия $|F(x) - P_m(x)| \le \varepsilon$ при всех $x \in [a, b]$ на основе оценки остаточного члена (нетривиальная задача численной оценки производных высших порядков), мы не гарантированы от того, что существует аппроксимирующий многочлен меньшей степени.

В основе методов равномерной аппроксимации лежит *теорема Чебышева*, утверждающая, что для существования многочлена $P_m(x)$ наилучшего приближения непрерывной функции $F(x)$ необходимо и достаточно существование на отрезке $[a, b]$ последовательности хотя бы $N = m + 2$ точек $x_0 < x_1 < x_2 < \ldots < x_m < x_{m+1}$ таких, что

$$F(x_i) - P_m(x_i) = \alpha\,(-1)^i\,L,\ i = 0, 1, \ldots, m + 1, \qquad (5.30)$$

где $\alpha = 1$ или -1, $L = \sup\limits_{x \in [a, b]} |F(x) - P_m(x)|$ (на практике вместо точной верхней границы sup выбирается максимальное из отклонений в указанных точках).

Для иллюстрации последовательности поиска многочлена возьмем функцию $F(x)$ вогнутую (или выпуклую) на $[a, b]$ и многочлен первой степени $P_1(x) = A + B\,x$. Примем за L максимальное из отклонений $|F(x) - P_m(x)|$ на $[a, b]$ и за x^* − точку максимума (минимума) разности $F(x) - P_1(x)$. Тогда возникает система трех уравнений

$$F(a) - A - B\,a = \alpha\,L,$$
$$F(x^*) - A - B\,x^* = -\alpha\,L,$$
$$F(b) - A - B\,b = \alpha\,L.$$

Из первого и третьего уравнений с очевидностью получаем $B = (F(b) - F(a)) / (b - a)$. Учитывая, что точка x^* является точкой экстремума $F(x) - P_1(x)$, получаем уравнение $F'(x^*) - B = 0$, решение которого дает искомую точку и возможность нахождения A и значения $\alpha\,L$.

В геометрической интерпретации (рис. 5.8) этот процесс сводится к построению хорды, связывающей точки $(a, F(a))$ и $(b, F(b))$, построению параллельной ей касательной к кривой

$y = F(x^*)$ и проведению прямой, равноудаленной от упомянутых хорды и касательной.

Для многочлена m-й степени возникает система $m + 2$ уравнений

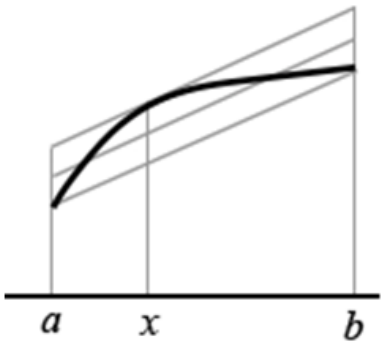


Рис. 5.8

$F(x_i) - (A_0 + A_1 x_i + \ldots + A_m x_i^m) = \alpha\,(-1)^i L$ ($i = 0, 1, \ldots, m + 1$) с $2m + 2$ неизвестными A_i ($i = 0, 1, 2, \ldots, m$), x_i ($i = 1, 2, \ldots, m$) и αL. Если учесть, что искомые значения x_i ($i = 1, 2, \ldots, m$) определяют экстремумы отклонений и соблюдаются некоторые условия гладкости функции, то эту систему можно дополнить еще m условиями:

$$F'(x_i) - (A_1 + 2 A_2 x_i + \ldots + m A_m x_i^{m-1}) = 0,\ i = 1, 2, \ldots, m.$$

Тем не менее, решение такой нелинейной системы для массового использования (для произвольной функции) едва ли реально.

Другой подход к поиску наилучшего равномерного приближения связан с построением *интерполяционного* многочлена m-й степени (см. ниже) с узлами интерполяции (*чебышевским альтернансом*)

$$x_k = \tfrac{a+b}{2} + \tfrac{b-a}{2} \cos\left(\tfrac{2k-1}{2(m+1)}\pi\right),\ k = \overline{1, m+1} \qquad (5.31)$$

(здесь концы интервала в число узлов не включаются; в узлах интерполяции отклонение полинома от исходной функции отсутствует).

Равномерные приближения обычно используются при создании стандартных вычислительных процедур для поиска значений часто используемых функций с целью минимизации времени вычислений. Так вместо аппроксимации $F(x) = \mathrm{arctg}(x)$ отрезком ряда Тейлора

$$\mathrm{arctg}(x) \cong x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \frac{x^{11}}{11}$$

можно взять эквивалентную по точности аппроксимацию многочленом пятого порядка

$$\mathrm{arctg}(x) \cong 0.9999374\, x - 0.3303433\, x^3 + 0.1632823\, x^5.$$

В случае табличных функций равномерные аппроксимации требуют от решающего задачу нестандартного подхода и колоссальных затрат энергии.

5.3. Интерполирование функций

Пусть функция $F(x)$ задана в точках $x_0, x_1, \ldots, x_n$ значениями $f_0, f_1, \ldots, f_n$. Задача интерполяции состоит в отыскании функции $\varphi(x)$, совпадающей с $F(x)$ в указанных $n + 1$ точках (узлах интерполяции)

$$\varphi(x_i) = F(x_i),\ i = 0, 1, \ldots, n. \tag{5.32}$$

Возьмем в качестве функции $\varphi(x)$ алгебраический полином n-й степени

$$P_n(x) = \sum_{k=0}^{n} a_k x^{n-k}. \tag{5.33}$$

Подставляя формулу (5.33) в (5.32), получаем систему уравнений, решение которой единственно (если все узлы различны) и реализует поставленную задачу ($n + 1$ точка однозначно определяет «параболу» n-го порядка):

$$\begin{aligned} &a_0 x_0^n + a_1 x_0^{n-1} + \ldots + a_n = f_0, \\ &a_0 x_1^n + a_1 x_1^{n-1} + \ldots + a_n = f_1, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_0 x_n^n + a_1 x_n^{n-1} + \ldots + a_n = f_n. \end{aligned} \tag{5.34}$$

Использование (5.34) является простейшим, но достаточно трудоемким методом построения интерполяционного многочлена.

Рассмотрим некоторые другие приемы интерполяции.

5.3.1. Интерполяционный многочлен Лагранжа

Интерполяционный многочлен Лагранжа имеет вид

$$L_n(x) = \sum_{k=0}^{n} f_k \frac{(x - x_0)(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)}{(x_k - x_0)(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)} \tag{5.35}$$

или в компактной форме

$$L_n(x) = \sum_{k=0}^{n} \left(f_k \prod_{i \neq k} \frac{x - x_i}{x_k - x_i} \right). \qquad (5.35')$$

В случае равноотстоящих узлов $x_k = x_0 + k\,h$ значения x можно представить в виде $x = x_0 + t\,h$ и многочлен Лагранжа записать как

$$L_n(x) = t\,(t-1)\ldots(t-n) \sum_{k=0}^{n} \frac{(-1)^{n-k}}{k!(n-k)!} \frac{f_k}{t-k}. \qquad (5.35'')$$

Представления (5.35) приемлемы при одиночных вычислениях, массовое же их использование трудоемко, кроме случаев линейной

$$L_1(x) = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0} = A + B\,x, \qquad (5.36)$$

где

$$A = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0}; \qquad B = \frac{f_1 - f_0}{x_1 - x_0},$$

и квадратичной интерполяции

$$L_2(x) = f_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}. \qquad (5.37)$$

Для функций, имеющих в заданном диапазоне непрерывные производные до $(n + 1)$-го порядка, остаточный член интерполяционного многочлена (погрешность аппроксимации) не превышает

$$R_n(f) = \max_{z \in [a,b]} F^{(n+1)}(z) \frac{1}{(n+1)!} \prod_{i=0}^{n} (x - x_i). \qquad (5.38)$$

Полезно иметь в виду, что интерполяционный многочлен Лагранжа *в случае монотонной функции* можно использовать и для решения *обратной задачи интерполяции*: поиска аргумента для заданного значения функции (**!**)

$$x \cong L_n(f) = \sum_{k=0}^{n} \left(x_k \prod_{i \neq k} \frac{f - f_i}{f_k - f_i} \right). \qquad (5.39)$$

В случае немонотонной функции, для которой обратная функция неоднозначна, приходится строить интерполяционный многочлен и решать уравнение $L_n(x) = f$, что при $N > 2$ не совсем приятно.

5.3.2. Конечные разности

Пусть имеется таблица значений f_k функции $F(x)$ для *равноотстоящих* значений аргумента $x_k = x_0 + k\,h$ $(k = 0, 1, 2, \ldots, N)$. Величины

$$\Delta f_k = f_{k+1} - f_k,\ k = 0, 1, \ldots, N - 1 \qquad (5.40)$$

называются конечными разностями первого порядка. Величины

$$\Delta^2 f_k = \Delta f_{k+1} - \Delta f_k = f_{k+2} - 2\,f_{k+1} + f_k,\ k = 0, 1, \ldots, N - 2 \qquad (5.41)$$

– конечными разностями второго порядка и т. д.

Обычно конечные разности записывают в виде таблиц. Например:

x_k	f_k	Δf_k	$\Delta^2 f_k$	$\Delta^3 f_k$	$\Delta^4 f_k$
0	1				
		2			
1	3		4		
		6		0	
2	9		4		0
		10		0	
3	19		4		
		14			
4	33				

С помощью представлений через разложение по Тейлору можно показать, что *отношение $\Delta^m f_k / h^k$ может быть принято за оценку m-й производной функции в точке x_k*. Например, $f''(x_k) = \Delta^2 f_k / h^2 + O(h)$, где $O(h)$ − величина порядка малости h^2.

x_k	f_k	Δf_k	$\Delta^2 f_k$
–1	**3**		
		–2	
0	1		4
		2	
1	3		4
		6	
2	9		4
		10	
3	19		4
		14	
4	33		4
		18	
5	**51**		
		22	
6	**73**		

Рис. 5.9

Очевидно, что для многочлена m-го порядка конечные разности m-го порядка должны быть одинаковыми, а порядка $m + 1$ и выше − нулевыми. Так в приведенном примере видим постоянство третьих разностей, откуда следует, что на интервале $(0, 4)$ имеет место интерполяционный многочлен третьего порядка.

Эти соображения могут быть положены в основу табличной экстраполяции (расширения за пределы таблицы). Так, обнаружив в приведенной таблице для значений $x_k = 0, 1, 2, 3, 4$ постоянство вторых разностей, обратным ходом пополняем экстраполяцию за пределы исходного диапазона (рис. 5.9).

При практическом использовании конечных разностей следует учитывать быстрый рост погрешности. Пусть в одном из значений функции допущена погрешность порядка ε. Приведенная ниже таблица показывает, что конечная разность 8-го порядка содержит ошибку, в 70 раз большую.

f_k	Δf_k	$\Delta^2 f_k$	$\Delta^3 f_k$	$\Delta^4 f_k$	$\Delta^5 f_k$	$\Delta^6 f_k$	$\Delta^7 f_k$	$\Delta^8 f_k$
0	0							
0	0	0						
0	0	0	0					
0	ε	ε	ε	ε				
ε	−ε	−2 ε	−3 ε	−4 ε	−5 ε			
0	0	ε	3 ε	6 ε	10 ε	15 ε		
0	0	0	−ε	−4 ε	−10 ε	−20 ε	−35 ε	
0	0	0	0	ε	5 ε	15 ε	35 ε	70 ε

Поэтому пользоваться разностями высших порядков нужно весьма осторожно, особенно при возможности ошибок в исходной таблице.

Наряду с приведенными выше конечными разностями используются и так называемые *центральные разности*, первая из которых определяется формулой $\Delta f_k = f_{k+1} - f_{k-1}$.

5.3.3. Интерполяционные формулы

Интерполяционные формулы (вывод их можно обнаружить практически во всех руководствах по численному анализу) позволяют отыскивать значения табличной функции в точках, отличных от узлов таблицы, *без построения интерполяционного многочлена*.

Самыми популярными из них (в том числе в компьютерных реализациях) являются интерполяционные формулы Ньютона.

Формула Ньютона для интерполирования вперед

$$f(x_k + t\,h) = f_k + t\,\Delta f_k + \frac{t(t-1)}{2!}\Delta^2 f_k + \ldots + \\ + \frac{t(t-1)\ldots(t-m+1)}{m!}\Delta^m f_k + \ldots \qquad (5.42)$$

удобна для использования в диапазоне узлов, удаленных от конца таблицы. Узел x_k подбирают для конкретного значения x так, чтобы величина $t = (x - x_k) / h$ принимала значения в диапазоне от 0 до 1.

Формула Ньютона для интерполирования назад

$$f(x_k - t\,h) = f_k - t\,\Delta f_{k-1} + \frac{t(t-1)}{2!}\Delta^2 f_{k-2} - \frac{t(t-1)(t-2)}{3!}\Delta^3 f_{k-3} + \\ + \ldots + (-1)^m \frac{t(t-1)\ldots(t-m+1)}{m!}\Delta^m f_{k-m} + \ldots \qquad (5.43)$$

удобна для использования в диапазоне узлов, удаленных от начала таблицы. Узел x_k подбирают для конкретного значения x так, чтобы величина $t = (x_k - x) / h$ принимала значения в диапазоне от 0 до 1.

Пример. Пусть задана функция и ее конечные разности [2].

x_k	f_k	Δf_k	$\Delta^2 f_k$	$\Delta^3 f_k$	$\Delta^4 f_k$
0.1	0.09983	0.09884	–0.00199	–0.00096	0.00002
0.2	0.19867	0.09685	–0.00295	–0.00094	
0.3	0.29552	0.09390	–0.00389		
0.4	0.38942	0.09001			
0.5	0.47943				

При поиске $f(0.14)$ разумнее выбрать начальный узел $x = 0.1$ ($h = 0.4$) и воспользоваться интерполированием вперед:

$$f(0.14) = 0.09983 + t0.09884 - \frac{t(t-1)}{2}0.00199 - \frac{t(t-1)(t-2)}{3!}0.00096 + \\ + \frac{t(t-1)(t-2)(t-3)}{4!}0.00002 = 0.13964.$$

При поиске $f(0.45)$ разумнее выбрать начальный узел $x = 0.5$ ($h = 0.5$) и воспользоваться интерполированием назад:

$$f(0.45) = 0.47943 - t\,0.09001 - \frac{t(t-1)}{2}0.00389 + \frac{t(t-1)(t-2)}{3!}0.00094 +$$

$$+ \frac{t(t-1)(t-2)(t-3)}{4!}0.00002 = 0.44396.$$

При интерполировании в середине таблиц можно пользоваться и другими интерполяционными формулами, которые строятся на основе конечных разностей, последовательно выбираемых из выделенных клеток приведенной таблицы.

x_{k-3}	f_{k-3}	Δf_{k-3}	$\Delta^2 f_{k-3}$	$\Delta^3 f_{k-3}$	$\Delta^4 f_{k-3}$	$\Delta^5 f_{k-3}$
x_{k-2}	f_{k-2}	Δf_{k-2}	$\Delta^2 f_{k-2}$	$\Delta^3 f_{k-2}$	$\Delta^4 f_{k-2}$	$\Delta^5 f_{k-2}$
x_{k-1}	f_{k-1}	Δf_{k-1}	$\Delta^2 f_{k-1}$	$\Delta^3 f_{k-1}$	$\Delta^4 f_{k-1}$	
x_k	f_k	Δf_k	$\Delta^2 f_k$	$\Delta^3 f_k$		
x_{k+1}	f_{k+1}	Δf_{k+1}	$\Delta^2 f_{k+1}$			
x_{k+2}	f_{k+2}	Δf_{k+2}				
x_{k+3}	f_{k+3}					

Примерами таких формул могут служить следующие представления, где $0 < t < 1$.

Интерполяционные формулы Гаусса:

1)
$$f(x_k + th) = f_k + t\Delta f_k + \frac{t(t-1)}{21}\Delta^2 f_{k-1} + \frac{t(t-1)(t+1)}{3!}\Delta^3 f_{k-1} +$$
$$+ \frac{t(t-1)(t+1)(t-2)}{4!}\Delta^4 f_{k-2} + \frac{t(t-1)(t+1)(t-2)(t+2)}{5!}\Delta^5 f_{k-2} + \ldots \qquad (5.44)$$

2)
$$f(x_k + th) = f_k + t\Delta f_{k-1} + \frac{t(t+1)}{21}\Delta^2 f_{k-1} + \frac{t(t+1)(t-1)}{3!}\Delta^3 f_{k-2} +$$
$$+ \frac{t(t+1)(t-1)(t+2)}{4!}\Delta^4 f_{k-2} + \frac{t(t+1)(t-1)(t+2)(t-2)}{5!}\Delta^5 f_{k-3} + \ldots \qquad (5.45)$$

Интерполяционная формула Стирлинга:

$$f(x_k + th) = f_k + t\frac{\Delta f_k + \Delta f_{k-1}}{2} + \frac{t^2}{2!}\Delta^2 f_{k-1} + \frac{t(t^2-1)}{3!} \times$$
$$\times \frac{\Delta^3 f_{k-1} + \Delta^3 f_{k-2}}{2} + \frac{t^2(t^2-1)}{4!}\Delta^4 f_{k-2} + \frac{t(t^2-1)(t^2-2^2)}{5!} \cdot \frac{\Delta^5 f_{k-3} + \Delta f_{k-2}}{2} + \ldots \qquad (5.46)$$

Интерполяционная формула Бесселя:

$$f(x_k + t\,h) = \frac{f_k + f_{k-1}}{2} + (t-0.5)\Delta\, f_k + \frac{t(t-1)}{2!}\frac{\Delta^2 f_k + \Delta^2 f_{k-1}}{2} +$$
$$+ \frac{t(t-1)(t-0.5)}{3!}\Delta^3 f_{k-1} + \frac{t(t^2-1)}{3!}\frac{\Delta^3 f_{k-1} + \Delta^3 f_{k-2}}{2} + \qquad (5.47)$$
$$+ \frac{(t+1)t(t-1)(t-2)}{4!}\frac{\Delta^4 f_{k-1} + \Delta^4 f_{k-2}}{2} + \frac{(t+1)t(t-1)(t-2)(t-0.5)}{5!}\Delta^5 f_{k-2} + \ldots$$

5.3.4. Интерполирование функций двух переменных

Пусть задана таблица значений функции двух переменных $z = f(x, y)$ и требуется найти $z = f(x^*, y^*)$. Эта задача решается в два этапа: сначала для всех табличных значений y_k проводится интерполяция по x для указанного x^* и затем по найденной таблице значений $f(x^*, y_k)$ проводится интерполяция по y для заданного y^*.

Высказанный прием распространяется и на случай функции многих переменных в предположении ее гладкости.

5.3.5. Численное дифференцирование

Задача численного дифференцирования возникает в случае, когда функция имеет исходное табличное задание или приведена к таковому по каким-то соображениям. Большинство формул численного дифференцирования получается дифференцированием интерполяционных формул. Так, получив интерполяционный многочлен Лагранжа, обычным дифференцированием можно, в принципе, получить его производные любого порядка.

Дифференцированием формулы Ньютона интерполирования вперед:

$$f(x_k + t\,h) = f_k + t\,\Delta f_k + \frac{t(t-1)}{2!}\Delta^2 f_k + \ldots + \frac{t(t-1)\ldots(t-m+1)}{m!}\Delta^m f_k + \ldots;$$

$t = \frac{x - x_k}{h}$ можно получить оценки производных вне узлов таблицы:

$$\frac{d}{dx} f(x_k + t\,h) = \frac{1}{h}\left[\Delta f_k + \frac{2t-1}{2!}\Delta^2 f_k + \frac{3t^2 - 6t + 2)}{3!}\Delta^3 f_k + \ldots\right]$$

и, фиксируя $t = 0$, в узлах:

$$f'(x_k) = \tfrac{1}{h}\left[\Delta f_k - \tfrac{1}{2}\Delta^2 f_k - \tfrac{1}{3}\Delta^3 f_k + \tfrac{1}{4}\Delta^4 f_k - \ldots\right],$$
$$f''(x_k) = \tfrac{1}{h^2}\left[\Delta^2 f_k - \Delta^3 f_k + \tfrac{11}{12}\Delta^4 f_k - \tfrac{5}{6}\Delta^5 f_k + \ldots\right], \qquad (5.48)$$
$$f'''(x_k) = \tfrac{1}{h^3}\left[\Delta^3 f_k - \tfrac{3}{2}\Delta^4 f_k + \tfrac{7}{4}\Delta^5 f_k - \ldots\right].$$

На практике редко учитывают разности высоких порядков и используют небольшое число слагаемых.

Если найти первые члены разложения функции в ряд Тейлора, то можно получить простые и эффективные в приложениях оценки:

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_k)}{h} + O(h),\ f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{h} + O(h), \qquad (5.49)$$

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_{k-1})}{2h} + O(h^2),$$

$$f'(x_k) = \frac{-3f(x_k) + 4f(x_{k+1}) - f(x_{k+2})}{2h} + O(h^2), \qquad (5.49')$$

$$f'(x_k) = \frac{f(x_{k-2}) - 4f(x_{k-1}) + f(x_k)}{2h} + O(h^2),$$

$$f''(x_k) = \frac{f(x_{k+1}) - 2f(x_k) + f(x_{k-1})}{h^2} + O(h^2). \qquad (5.50)$$

Читатель может установить приведенные оценки, подставив сюда разложения функции в ряд Тейлора в точках, смежных с x_k. Можно построить множество подобных и более точных формул численной оценки производных для табличных функций (случай равноотстоящих узлов) при дифференцировании в середине и на концах таблицы.

Другой путь построения формул численного дифференцирования связан с *методом неопределенных коэффициентов*. Обозначим для краткости $f(x_k) = f_k$ и попытаемся выразить ее производную через $m + 1$ значение f_k:

$$f'(x_k) = A_0 f_0 + A_1 f_1 + \ldots + A_m f_m. \qquad (5.51)$$

Полагая, что эта формула точна для полинома m-й степени и, в частности, для полиномов

$$1,\ x - x_k,\ (x - x_k)^2,\ \ldots,\ (x - x_k)^m, \qquad (5.52)$$

подстановкой (5.52) в (5.51) получаем систему из $(m + 1)$-го уравнения для вычисления неопределенных коэффициентов.

Например, выразим производную функции в точке x_1 через значения f_0, f_1, f_2, f_3:

$$f'(x_1) = A_0 f_0 + A_1 f_1 + A_2 f_2 + A_3 f_3.$$

Подставляя соответствующие степени $x - x_k$, с учетом $x_k = x_0 + k\,h$, получаем систему

$$\begin{aligned}
0 &= A_0 + A_1 + A_2 + A_3,\\
1 &= A_1 h + A_2 (2\,h) + A_3\, 3\,h,\\
2\,h &= A_1 h^2 + A_2 (2\,h)^2 + A_3 (3\,h)^2,\\
3\,h^2 &= A_1 h^3 + A_2 (2\,h)^3 + A_3 (3\,h)^3,
\end{aligned}$$

решение которой дает

$$f'(x_1) = [-2 f_0 - 3 f_1 + 6 f_2 - f_3] / (6\,h).$$

Для оценки качества полученного результата возьмем разложение функции в ряд Тейлора в окрестности точки x_1

$$f_0 = f(x_1 - h) = f_1 - h\, f_1' + \frac{h^2}{2!} f_1'' - \frac{h^3}{3!} f_1''' + \frac{h^4}{4!} f_1^{(4)} + \ldots,$$

$$f_2 = f(x_1 + h) = f_1 + h\, f_1' + \frac{h^2}{2!} f_1'' + \frac{h^3}{3!} f_1''' + \frac{h^4}{4!} f_1^{(4)} + \ldots,$$

$$f_3 = f(x_1 + 2h) = f_1 + 2h\, f_1' + \frac{4h^2}{2!} f_1'' + \frac{8h^3}{3!} f_1''' + \frac{16h^4}{4!} f_1^{(4)} + \ldots,$$

получая

$$f'(x_1) = f_1' - f_1^{(4)}\, h^3 / 12,$$

т. е. полученная формула имеет ошибку порядка $O(h^3)$.

Как мы увидим далее, приведенные формулы численного дифференцирования находят применение как в задачах решения нелинейных уравнений при оценке условий сходимости, так и в численных методах решения дифференциальных уравнений.

Здесь мы ограничились только случаем равноотстоящих узлов. В общем случае можно пользоваться дифференцированием многочлена Лагранжа или интерполяционных формул для неравноотстоящих узлов, но этот путь не привлекателен из-за громоздких преобразований. Подчас разумнее найти значения интерполяционного многочлена на равномерной сетке (это проще поиска его коэффициентов при наличии компьютера) и пользоваться вышеприведенными формулами.

5.4. Интерполирование сплайнами

Рассмотренные выше методы интерполирования при использовании большого числа узлов часто дают слишком большую вычислительную погрешность, и к тому же многочлены высоких порядков не слишком удобны. Поэтому для уменьшения погрешности предпочтительно область интерполяции разбить на несколько подынтервалов и на каждом из них использовать для аппроксимации полином невысокой степени, т. е. воспользоваться *кусочно-полиномиальной аппроксимацией*.

Один из способов такой аппроксимации связан с использованием интерполяции сплайнами. Пусть отрезок $[a, b]$ разбит на N подынтервалов с граничными узлами

$$a = x_0 < x_1 < \ldots < x_{N-1} < x_N = b.$$

Сплайном m-го порядка для $f(x)$ на $[a, b]$ называют кусочную функцию $P(x) = \{P_1(x), P_2(x), \ldots, P_N(x)\}$, удовлетворяющую условиям:

1) все функции $P_k(x)$ $(k = 1, 2, \ldots, N)$ являются полиномами m-го порядка;

2) для концов подынтервалов соблюдаются условия интерполяции и непрерывности

$$P_1(x_0) = f(x_0),\ P_N(x_N) = f(x_N),\ P(x_k) = P_{k+1}(x_k) = f(x_k), \qquad (5.53)$$

где $k = 1, 2, \ldots, N - 1$;

3) на концах подынтервалов соблюдаются условия непрерывности производных до $(m - 1)$-го порядка

$$P_k^{(s)}(x_k) = P_{k+1}^{(s)}(x_k),\ k = 1, 2, \ldots, N-1,\ s = 1, 2, \ldots, m - 1. \qquad (5.54)$$

На рис. 5.10 приведен сплайн, в котором значения функций

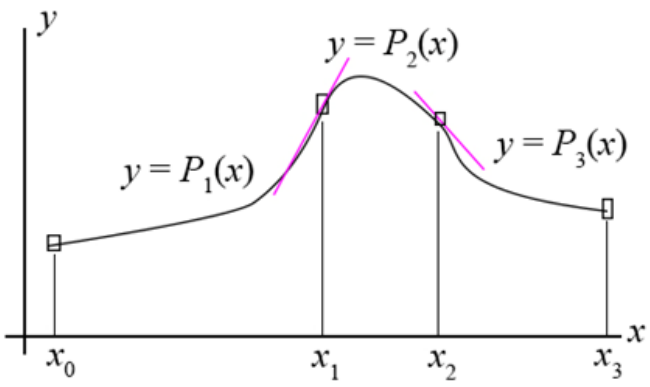


Рис. 5.10

в выбранных узлах совпадают (условие интерполяции) и наблюдается гладкий переход от одной кривой к другой – равенство производных слева и справа.

Если взять простейшую задачу − аппроксимацию линейными сплайнами ($m = 1$)

$$P_k(x) = A_k\, x + B_k,\ k = 1, 2, \ldots, N, \qquad (5.55)$$

то из (5.53) возникает система $2N$ уравнений с $2N$ неизвестными ($N + 1$ условие интерполяции и $N - 1$ условие непрерывности):

$$A_1 x_0 + B_1 = f(x_0),\ A_k\, x_k + B_k = f(x_k),\ k = 1, 2, \ldots, N, \qquad (5.56)$$
$$A_{k+1} x_k + B_{k+1} = f(x_k),\ k = 1, 2, \ldots, N - 1,$$

решение которой тривиально:

$$A_k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}},\ B_k = f(x_k) - A_k\, x_k\ ,\ k = 1, 2, \ldots, N. \qquad (5.57)$$

Линейные сплайны непопулярны – не всем нравятся эти ломаные линии, но приемлемы для чисто технических расчетов.

При аппроксимации квадратичными сплайнами

$$P_k(x) = A_k\,(x - x_{k-1})^2 + B_k\,(x - x_{k-1}) + C_k,\ k = 1, 2, \ldots, N, \qquad (5.58)$$

возникает система из $3N - 1$ уравнения с $3N$ неизвестными

$P_k(x_{k-1}) = f(x_{k-1}) \Rightarrow C_k = f(x_{k-1}),\ k = 1, 2, \ldots, N,$

$P_k(x_k) = f(x_k) \Rightarrow A_k\,(x_k - x_{k-1})^2 + B_k\,(x_k - x_{k-1}) + C_k = f(x_k),\ k = 1, 2, \ldots, N,$

$P'_k(x_k) = P'_{k+1}(x_k) \Rightarrow 2\,A_k\,(x_k - x_{k-1}) + B_k = B_{k+1},\ k = 1, 2, \ldots, N - 1.$

Недостающее уравнение получают, ставя какое-нибудь условие на одном из концов интервала, например, требуя задания производной $P'_1(x_0)$ равной нулю (?), откуда $B_1 = 0$, или прибегнув к какой-либо аппроксимации типа (5.49) при равноотстоящих узлах:

$$f'(x_0) = \frac{-3f(x_0) + 4f(x_1) - f(x_2)}{2h} + O(h^2)\ .$$

Соответственно:

$$C_k = f(x_{k-1}), k = \overline{1, N},$$
$$B_1 = ?,\ B_{k+1} = 2\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} - B_k,\ k = \overline{1, N}, \qquad (5.59)$$
$$A_k = \frac{f(x_k) - f(x_{k-1}) - B_k(x_k - x_{k-1})}{(x_k - x_{k-1})^2},\ k = \overline{1, N}.$$

Аналогично можно получить параметры кубической сплайн-интерполяции, где требуется выбор уже двух констант (в среде MatLab реализуется командой `pp=spline(x,f)`). Например, задав `x=0:10` и `f=sin(x)*exp(-x)` или любому массиву из 11 значений, мы получаем возможность *найти значения* кубического сплайна в любых точках диапазона [0, 10] (рис. 5.11):

```
xx=0:.25:10; yy=spline(x,y,xx);
plot(x,y,'o',xx,yy)
```

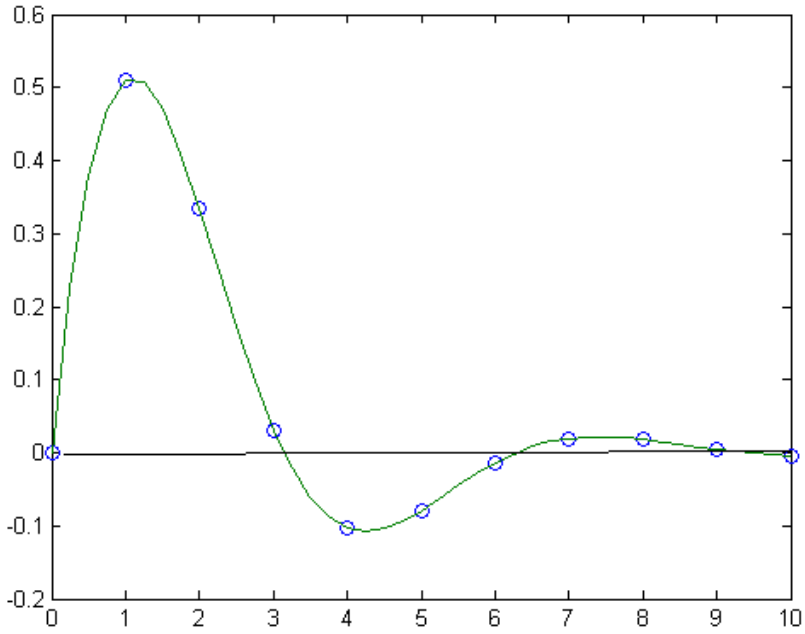


Рис. 5.11

Несмотря на популярность идеологии сплайнов в последние годы, сплайны более высокого порядка практически никто не использует.

Глава 6. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Задача вычисления определенного интеграла, по существу, появилась уже при возникновении частной собственности: как поделить («по справедливости», конечно) пахотную землю или пастбище между отдельными представителями племени? Как найти площадь угодья? Для фигур типа прямоугольника и трапеции проблему решили достаточно давно. То ли Архимед, то ли Герон Александрийский предложили популярную до наших дней формулу для расчета площади треугольника по длинам его сторон. Многие столетия решалась задача о квадратуре круга до тех пор, пока не появилось число π (это обозначение введено Л. Эйлером в 1737 г.).

Создание дифференциального и интегрального исчисления (И. Ньютон и Г. Лейбниц) стало своеобразной революцией в математике, обобщив результаты исследований многих веков. Сегодня даже школьник знает, что если для функции $f(x)$, определенной на $[a, b]$, удается *найти первообразную* $F(x)$ (взять *интеграл*), то значение определенного интеграла от этой функции с легкостью определяется по формуле Ньютона – Лейбница:

$$\int_a^b f(x)dx = F(b) - F(a). \qquad (6.1)$$

Но попробуйте *взять интеграл* от достаточно простой функции $\frac{\sin(x)}{x}e^{-x}$ или отыскать пороговое значение x, при котором вероятность выхода некой случайной величины t за порог не превысит значения α (например, 0.95 или 0.9) $\frac{1}{\sigma\sqrt{2\pi}}\int_0^x e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt = \alpha$. Разумеется, для последней из задач, возникающей при математической обработке разнообразных данных, за последние 150 лет созданы таблицы так называемого нормального распределения (распределения Гаусса) вероятностей, но в компьютерный век их использование не всегда удобно.

Для большинства аналитически заданных функций, не говоря уже о табличных функциях, поиск первообразной нереален, и потому возникает задача численной оценки вероятности с какой-то точностью.

6.1. Квадратурные формулы Ньютона – Котеса

Построение *квадратурных формул* − формул численного интегрирования функции одной переменной − базируется на замене исходной функции аппроксимирующей функцией, интеграл от которой легко берется. Так для функции $f(x)$, вычисленной в каких-то точках − узлах $x_0, x_1, \ldots, x_m$ отрезка $[a, b]$, можно построить интерполяционный многочлен Лагранжа

$$L_m(x) = \sum_{k=0}^{m} f_k \prod_{i \neq k} \frac{x - x_i}{x_k - x_i}$$

и в результате интегрирования получить квадратурную формулу

$$\int_a^b f(x)dx = \sum_{k=0}^{m} A_k \; f_k + R_m[f], \qquad (6.2)$$

где $R_m[f]$ − ошибка квадратур и

$$A_k = \int_a^b \prod_{i \neq k} \frac{x - x_i}{x_k - x_i}. \qquad (6.3)$$

Однако хотелось бы обойтись без интегрирования многочлена Лагранжа. Заметим, что для многочлена степени не выше m квадратура (6.2) является точной, т. е. $R_m[f] = 0$. Соответственно, полагая $f(x) = x^L$ ($L = 0, 1, \ldots, m$), подстановкой в (6.2) получаем систему линейных уравнений для определения коэффициентов квадратурной формулы на выбранной сетке узлов

$$\sum_{k=0}^{m} A_k x_k^L = \frac{1}{L+1}(b^{L+1} - a^{L+1}), k = \overline{0, m}. \qquad (6.4)$$

Большинство известных квадратурных формул строится на задании системы равноотстоящих точек $a + k\,h$ ($k = 0, 1, \ldots, m$), где $h = (b - a) / m$. Если пределы интегрирования a и b входят в состав узлов квадратурной формулы (6.2), то ее называют *квадратурой открытого типа* (в противном случае − *замкнутого*

типа). Создаваемые таким образом (при принудительном выборе узлов) формулы называются формулами Ньютона – Котеса.

Создадим, например, формулу (6.4) *открытого* типа с тремя узлами $a + h$, $a + 2h$, $a + 3h$, где $h = (b - a) / 4$. Построив и решив систему

$$\begin{aligned} &A_1 + A_2 + A_3 = b - a, \\ &A_1 \tfrac{3a+b}{4} + A_2 \tfrac{a+b}{2} + A_3 \tfrac{a+3b}{4} = \tfrac{1}{2}(b^2 - a^2), \\ &A_1\left(\tfrac{3a+b}{4}\right)^2 + A_2\left(\tfrac{a+b}{2}\right)^2 + A_3\left(\tfrac{a+3b}{4}\right)^2 = \tfrac{1}{3}(b^3 - a^3), \end{aligned}$$

получаем формулу вида

$$\int_a^b f(x)dx \cong (b-a)\left[\tfrac{2}{3} f(\tfrac{3a+b}{4}) - \tfrac{1}{3} f(\tfrac{a+b}{2}) + \tfrac{2}{3} f(\tfrac{a+3b}{4})\right]. \quad (6.5)$$

Создадим формулу *закрытого* типа с тремя узлами a, $(a + b) / 2$, b. Построив и решив систему

$$\begin{aligned} &A_1 + A_2 + A_3 = b - a, \\ &A_1 \cdot a + A_2 \cdot \tfrac{a+b}{2} + A_3 \cdot b = \tfrac{1}{2}(b^2 - a^2), \\ &A_1 \cdot a^2 + A_2 \cdot \left(\tfrac{a+b}{2}\right)^2 + A_3 \cdot b^2 = \tfrac{1}{3}(b^3 - a^3), \end{aligned}$$

получаем популярную *формулу парабол* (*формулу Симпсона*)

$$\int_a^b f(x)dx \cong \tfrac{b-a}{6}\left[f(a) + 4 f(\tfrac{a+b}{2}) + f(b)\right]. \quad (6.6)$$

Если ограничиться одним узлом $x = (a + b) / 2$ (центральным узлом), то получим *формулу прямоугольников с центральным узлом*

$$\int_a^b f(x)dx \cong (b-a)\, f(\tfrac{a+b}{2}). \quad (6.7)$$

Если взять два узла a и b, получаем *формулу трапеций*

$$\int_a^b f(x)dx \cong \tfrac{b-a}{2}\left[f(a) + f(b)\right]. \quad (6.8)$$

Можно построить множество подобных формул, но без суждения о величине остаточного члена это будет искусством ради искусства.

Возьмем формулу прямоугольников (6.7) и попытаемся оценить ее остаточный член (порядок погрешности)

$$R[h] = \int_a^{a+h} f(x)dx - h\, f(a+\tfrac{h}{2}).$$

Дифференцируя по h, получим

$$R'(h) = f(a+h) - f(a+\tfrac{h}{2}) - \tfrac{h}{2} f'(a+\tfrac{h}{2}) =$$

$$= f(a+\tfrac{h}{2}) + \tfrac{h}{2} f'(a+\tfrac{h}{2}) + \tfrac{h^2}{8} f''(a+\tfrac{h}{2}) + \ldots$$

$$\ldots - f(a+\tfrac{h}{2}) - \tfrac{h}{2} f'(a+\tfrac{h}{2}) = \tfrac{h^2}{8} f''(a+\tfrac{h}{2}) + \ldots$$

Интегрируя полученную оценку по h, получаем

$$R(h) \approx \int_0^h \tfrac{h^2}{8} f''(a+\tfrac{h}{2})dh \approx f''(c)\int_0^h \tfrac{h^2}{8} dh = \tfrac{h^3}{24} f''(c),$$

где c – некоторая точка из отрезка $[a, b]$.

Аналогичным образом можно получить оценки для других упомянутых формул [2]. Так для формулы трапеций

$$R[f] = -\frac{h^3}{12} f''(c), \quad h=b-a, \qquad c \in [a,b]. \tag{6.8′}$$

Для формулы Симпсона

$$R[f] = -\frac{h^5}{90} f^{(4)}(c), \quad h=\frac{b-a}{2}, \qquad c \in [a,b]. \tag{6.6′}$$

Среди других квадратур Ньютона – Котеса с равномерной сеткой [2] узлов интегрирования можно выделить следующие, относительно простые по форме и часто встречающиеся в приложениях.

Четырехточечная формула Ньютона – Котеса:

$$\int_a^b f(x)dx \cong \tfrac{b-a}{8}\left[f(a) + 3f(\tfrac{3a+b}{4}) + 3f(\tfrac{a+3b}{4}) + f(b)\right],$$

$$R[f] = -\frac{3\,h^5}{80} f^{(4)}(c)\ ,\ h = \frac{b-a}{3}. \tag{6.9}$$

Формула Боде (*пятиточечная формула замкнутого типа*):

$$\int_a^b f(x)dx \cong \frac{2h}{45}\left[7f(a)+32f(a+h)+12f(\frac{a+b}{2})+32f(b-h)+7f(b)\right], \quad (6.10)$$

$$R[f]=-\frac{8h^7}{945}f^{(6)}(c),\ h=\frac{b-a}{4}\ .$$

Семиточечная формула замкнутого типа:

$$\int_a^b f(x)dx \cong \frac{h}{140}[41f(a)+216f(a+h)+27f(a+2h)+\ 272f(\frac{a+b}{2})+$$

$$+27f(b-2h)+216f(b-h)+41f(b)], \quad (6.11)$$

$$R[f]=-\frac{9h^9}{1400}f^{(4)}(c),\quad h=\frac{b-a}{6}\ .$$

Формула Веддля (получается из предыдущей округлением коэффициентов):

$$\int_a^b f(x)dx \cong \frac{3h}{10}[f(a)+5f(a+h)+\ f(a+2h)+$$

$$+6f(\frac{a+b}{2})+f(b-2h)+5f(b-h)+\ f(b)],\quad h=\frac{b-a}{6}\ . \quad (6.12)$$

Показано [2], что оценки остаточных членов квадратурных формул зависят от четности числа (n) узлов формулы и ее типа ($s = 0$ − закрытого и $s = 1$ − открытого):

$$R_n \cong h^{n+2}\ f^{(n+1)}(c) \text{ для нечетного } n;$$

$$R_n \cong h^{n+1}\ f^{(n)}(c) \text{ для четного } n$$

(преимущество формул с нечетным числом узлов очевидно).

Там же [2] заинтересованный читатель может обнаружить таблицы коэффициентов A_k формул Ньютона – Котеса открытого и закрытого типов (до10 узлов) и оценки их остаточных членов.

Обычная практика квадратур отнюдь не предполагает применения квадратурной формулы ко всему промежутку интегрирования. Если имеется оценка максимума соответствующей производной, то из выражения остаточного члена можно найти приемлемое значение шага h, разделить интервал на отрезки длины h и отыскать сумму квадратур по этим отрезкам. Соответственно получим следующие формулы:

1) прямоугольников (с центральным узлом)

$$\int_a^b f(x)dx \cong h\sum_{i=1}^{N} f\left(a+\frac{2i-1}{2}h\right),\ h=\frac{b-a}{N};$$

2) трапеций

$$\int_a^b f(x)dx \cong h\left[\frac{f(a)+f(b)}{2}+\sum_{i=1}^{N-1} f(a+ih)\right],\ h=\frac{b-a}{N};$$

3) парабол (Симпсона)

$$\int_a^b f(x)dx \cong \frac{h}{6}\left[f(a)+f(b)+4\sum_{i=1}^{N} f(a+\frac{2i-1}{2}h)+2\sum_{i=1}^{N-1} f(a+ih)\right],$$

$$h=\frac{b-a}{N}.$$

В реальности надежные оценки производных чаще всего найти не удается. Поэтому используют так называемую *схему двойного пересчета*: отыскивается оценка при некотором начальном шаге (например, равном длине интервала), затем берется *вдвое меньший* шаг и поиск оценки повторяется. Если результаты двух очередных оценок близки (в смысле абсолютной или относительной погрешности), то последняя из них принимается за итоговую, иначе берется еще вдвое меньший шаг и т. д. Есть, конечно, опасность впасть в ошибку при таком подходе. Так, при поиске интеграла от $e^{-x}\sin(x)$ на интервале от 0 до 2π применение формулы трапеций с начальным шагом $h = 2\pi$ дает совпадающие нулевые оценки, тогда как значение интеграла отлично от нуля.

При организации такого пересчета некоторые формулы, где узлы предыдущего шага составляют часть узлов последующего, можно, как минимум вдвое, уменьшить общее время вычислений. Примером может служить реализация формулы Симпсона в среде MatLab

```
I0=1e19; h=(b-a);
s0=f(a)+f(b); s2=0; I1=s0*h/2; n=1;
while abs(I0-I1)>eps
 I0=I1; s1=0;
 for i=1:n s1=s1+f(a+(2*i-1)*h/2); end
```

```
 I1=(s0+4*s1+2*s2)*h/6; s2=s2+s1; h=h/2; n=2*n;
end
Itog=I1
```

6.2. Квадратурные формулы Чебышева

П. Л. Чебышевым была предложена группа квадратурных формул с неравноотстоящими узлами, но одинаковым весом слагаемых

$$\int_{-1}^{1} F(t)dt = A\sum_{k=1}^{m} F(t_k). \qquad (6.13)$$

Очевидно, что по этой формуле можно вычислять любой интеграл сведением к указанному интервалу

$$\int_{a}^{b} f(x)dx = \frac{b-a}{2}\int_{-1}^{1} F(t)dt,\ F(t) = f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right). \qquad (6.14)$$

Требуя, чтобы эта формула была точной для многочленов степени n, в частности для 1, t, t^2, …, t^n, получим систему нелинейных уравнений

$$\begin{aligned}
&A\,n = 2,\\
&A[t_1 + t_2 + ...t_n] = 0,\\
&A[t_1^2 + t_2^2 + ... + t_n^2] = \frac{2}{3},\\
&A[t_1^3 + t_2^3 + ... + t_n^3] = 0,\\
&A[t_1^4 + t_2^4 + ... + t_n^4] = \frac{2}{5},\\
&\text{.....}\\
&A[t_1^n + t_2^n + ... + t_n^n] = \frac{1+(-1)^n}{n+1}.
\end{aligned} \qquad (6.15)$$

Очевидно, что $A = 2 / n$. Заметим, что приведенная система (6.15) имеет действительные решения только при $n \le 7$ и $n = 9$, причем соблюдается симметрия $t_k = t_{n-k+1}$ ($k = 1, 2, \ldots, n / 2$).

Остаточный член квадратурной формулы Чебышева

$$R_n[f] = \frac{p(n)}{q(n)} F^{(n+1)}(c), \qquad (6.16)$$

где $p(n)$ / $q(n)$ приведено [2] в таблице.

Таблица параметров квадратурной формулы Чебышева

n	A	t_1	t_2	t_3	t_4	t_5	$p(n)$ / $q(n)$
1	2	0					1 / 3
2	1	0.577350					1 / 135
3	2/3	0.707107	0				1 / 360
4	1/2	0.794654	0.187592				2 / 42525
5	2/5	0.832498	0.374541	0			13 / 544320
6	2/3	0.866247	0.422519	0.266635			1 / 3969000
7	2/7	0.883862	0.529657	0.323912	0		28 / 195955200
9	2/9	0.911589	0.601019	0.528762	0.167906	0	0.7415 / 11!

6.3. Квадратурные формулы Гаусса

Квадратурные формулы Гаусса отличаются от формул Ньютона – Котеса и Чебышева тем, что здесь подлежат определению и узлы интегрирования, и весовые коэффициенты. Как было замечено выше (6.14), всякий интеграл по конечному промежутку можно свести к интегралу по промежутку от −1 до 1:

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} F(t)dt,\ F(t) = f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right).$$

Возьмем квадратурную формулу

$$\int_{-1}^{1} F(t)dt = \sum_{k=1}^{m} C_k F(t_k) \qquad (6.17)$$

и потребуем, чтобы она была точной для многочленов до $(2n-1)$-го порядка (например, 1, t, t^2, ..., t^{2n-1}). Это требование порождает систему $2n$ нелинейных уравнений с $2n$ неизвестными, решение которой нетривиально даже с учетом симметрии узлов и коэффициентов $t_k = t_{n-k+1}$, $C_k = C_{n-k+1}$ ($k = 1, 2, \ldots, n/2$) (при $n = 1$ и $n = 2$ формула Гаусса совпадает с формулой Чебышева).

Остаточный член n-точечной формулы Гаусса имеет вид [2]

$$R_n[f] = \frac{(b-a)^{2n+1}(n!)^4}{(2n)!^3(2n+1)} f^{(2n)}(c),\ c \in [a,b] \qquad (6.18)$$

(сравните эту оценку с оценками для других квадратур).

Узлы и коэффициенты квадратурных формул и коэффициент R при производной в остаточном члене для $a = -1$, $b = 1$ собраны в приведенной таблице (здесь мы ограничиваемся лишь небольшими n, вполне достаточными для приложений).

Таблица параметров квадратурной формулы Гаусса [2]

n	t_k	C_k	R
1	0	2	1/3
2	0.577350269	2	1/135
3	0.774596669 5/9	0 8/9	1/15750
4	0.861136311 0.339981043	0.173927422 0.326072577	1/3472875
5	0.906179845 0.538469310 0	0.118463442 0.239314335 0.284444444	1/1237732650
6	0.932469514 0.661209386 0.238619186	0.085662246 0.180380768 0.233956976	1/648984486150
7	0.949107912 0.741531185 0.405845151 0	0.064742483 0.139852696 0.190915025 0.208979592	$0.213 \cdot 10^{-14}$

Более полный свод узлов и коэффициентов можно найти в [2].

Пример. Для иллюстрации эффекта использования различных квадратурных формул вычислим интеграл

$$\int_0^1 \frac{1}{1+x^2}\,dx$$

при числе узлов, равном 5.

Использование различных формул квадратур дает следующие результаты:

Формула	**Оценка интеграла**
прямоугольников	0.78623114
трапеций	0.78279412
Симпсона	0.78539216
Боде	0.78552941
Чебышёва	0.78539815
Гаусса	0.78539816

(точное значение равно 0.785398163).

Легко видеть, что формула прямоугольников имеет преимущество при монотонной функции. Формула трапеций дает завышенные оценки при выпуклой функции и заниженные при вогнутой.

Наряду с приведенными квадратурными формулами используются метод Ромберга [4], формулы Эрмита [2] для интегралов типа $\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}} dx$, формулы Маркова [2], отличающиеся от гауссовых включением концов интервала в число узлов интегрирования.

В упомянутых выше системах двойного пересчета при числе разбиений исходного интервала n и $2n$ прибегают к *экстраполяции по Ричардсону* [2, 4] и за оценку интеграла принимают не I_{2n}, а более точную

$$J = I_{2n} + \frac{1}{2^m - 1}(I_{2n} - I_n), \qquad (6.19)$$

где m − порядок остаточного члена соответствующей квадратурной формулы (2 − для прямоугольников и трапеций, 4 − для Симпсона, 8 − для четырехточечной формулы Гаусса).

Заметим, что иногда можно избежать применения квадратур для часто используемых функций путем разложения их в ряд Тейлора с последующим почленным интегрированием, замены аппроксимирующим многочленом и т. д. Так вычисление интеграла

$$\int_0^x \frac{\sin(t)}{t} dt = \int_0^x \frac{1}{t}\left[t - \frac{t^3}{3!} + \frac{t^5}{5!} - \ldots \right] dt = x - \frac{x^3}{3 \cdot 3!} + \frac{x^5}{5 \cdot 5!} + \ldots$$

сводится к вычислению суммы ряда. Вычисление интеграла веро-

ятностей (функции Лапласа) можно свести к вычислению многочлена

$$\sqrt{\frac{2}{\pi}}\int_0^x \mathrm{e}^{-\frac{t^2}{2}}\,dt = 1 - \left[1 + 10^{-6}\,x\,(C_6 + x\,(C_5 + x\,(C_4 + x\,(C_3 + x\,(C_2 + x\,C_1))))))\right]^{-16},$$

$$C_1 = 5.383,\ C_2 = 48.891,\ C_3 = 38.004,\ C_4 = 3277.626,$$

$$C_5 = 21141.006,\ C_6 = 49867.347$$

с погрешностью не свыше $5\cdot10^{-7}$.

Весьма сложную проблему представляет и интегрирование быстро осциллирующих функций.

6.4. Вычисление несобственных интегралов

Иногда приходится иметь дело с несобственными интегралами, т. е. с интегралами от неограниченной функции или от функции по неограниченной области интегрирования. Естественно, что здесь при рассмотрении предела интегральных сумм обнаруживается их сходимость или отсутствие таковой. Так интеграл по неограниченному промежутку

$$\int_a^\infty \frac{dx}{x^a}\ ,\ a > 0$$

расходится при $a \le 1$ и сходится при $a > 1$.

Но даже при уверенности в сходимости интеграла прямое применение квадратурной формулы нереально из-за необходимости как-то ограничить интервал интегрирования и затем долго дробить его на подынтервалы с надеждой добиться близости оценок. Поэтому к отдельным классам функций приходится подходить со своим «аршином». Так для интегралов типа

$$\int_0^\infty f(x)\sin(kx)dx,$$

где $f(x)$ − ограниченная знакопостоянная функция, при $x \to \infty$ стремящаяся к нулю быстрее чем $1 / x$, разумно найти нули синуса и заменить этот интеграл суммой интегралов

$$\sum_{i=0}^{\infty} \int_{\pi i/k}^{\pi(i+1)/k} f(x)\sin(k\,x)\,dx.$$

Вычисление каждого из интегралов можно вести обычным путем по облюбованной квадратурной формуле (с точностью, например, на порядок выше заданной) и перебирать эти интегралы до тех пор, пока оценка очередного интеграла (или отношение ее к сумме предыдущих интегралов) не окажется меньше заданной точности (это следует из теоремы − утверждения о том, что погрешность вычисления суммы знакочередующегося ряда не превышает величины отбрасываемого члена). Если для $f(x)$ отсутствует ограничение знакопостоянства, то такой подход иногда может привести к неверным результатам (по возможности найдите не только нули синуса, но и нули $f(x)$).

Если подынтегральная функция знакопостоянна, то можно зафиксировать постоянную длину подынтервалов и последовательно накапливать сумму соответствующих оценок интегралов до тех пор, пока не выполнятся условия по точности (обычно завышенной на порядок).

6.5. Кубатурные формулы

При вычислении двойных интегралов вместо термина «квадратурная формула» используется термин «*кубатурная формула*». Для построения кубатурной формулы берется сетка (равномерная или неравномерная) точек, покрывающая область интегрирования, и строится формула

$$\iint\limits_{(S)} f(x,y)\,dx\,dy \approx \sum_{i=1}^{n}\sum_{j=1}^{m_i} A_i B_j f(x_i,y_j),\ (x_i,y_j)\in S. \qquad (6.20)$$

Например, если область интегрирования есть прямоугольник $S = [a \le x \le b,\ c \le y \le d]$, то можно построить кубатурную формулу Симпсона

$$\iint\limits_{(S)} f(x,y)\,dx\,dy \approx \frac{(b-a)(d-c)}{4\cdot 9}\left[\begin{array}{c} f(a,c)+f(a,d)+f(b,c)+f(b,d)+ \\ +4\{f(a,\frac{c+d}{2})+f(b,\frac{c+d}{2})+f(\frac{a+b}{2},c)+ \\ +f(\frac{a+b}{2},d)\}+16f(\frac{a+b}{2},\frac{c+d}{2}) \end{array}\right].$$

Многоточечная (2 n разбиений по x и 2 m − по y) кубатурная формула Симпсона имеет вид

$$\iint\limits_{S} f(x,y)\,dx\,dy \approx \frac{(b-a)(d-c)}{9}\sum_{i=0}^{2n}\sum_{j=0}^{2m}\lambda_{ij}\,f(a+ih,c+jk), \qquad (6.21)$$
$$h=\frac{b-a}{2n},\ k=\frac{d-c}{2m}.$$

Ее коэффициенты есть элементы матрицы

$$\Lambda = \begin{array}{|cccccccccc|} \hline 1 & 4 & 2 & 4 & 2 & ... & 4 & 2 & 4 & 1 \\ 4 & 16 & 8 & 16 & 8 & ... & 16 & 8 & 16 & 4 \\ 2 & 8 & 4 & 8 & 4 & ... & 8 & 4 & 8 & 2 \\ ... & ... & ... & ... & .... & ... & ... & ... & ... & ... \\ 2 & 8 & 4 & 8 & 4 & ... & 8 & 4 & 8 & 2 \\ 4 & 16 & 8 & 16 & 8 & ... & 16 & 8 & 16 & 4 \\ 1 & 4 & 2 & 4 & 2 & ... & 4 & 2 & 4 & 1 \\ \hline \end{array}$$

Заметим, что для непрямоугольной области S можно воспользоваться той же формулой, если найти содержащий ее прямоугольник R и вместо интегрирования $f(x,y)$ интегрировать функцию

$$f^{*}(x,y)=\begin{cases} f(x,y), & (x,y)\in S \\ 0, & (x,y)\notin S \end{cases}. \qquad (6.22)$$

Можно построить и примитивную формулу прямоугольников

$$\iint\limits_{(R)} f(x,y)\,dx\,dy \approx (b-a)\,(d-c)\,f(\frac{a+b}{2},\frac{c+d}{2})$$

или

$$\iint\limits_{R} f(x,y)\,dx\,dy \approx hk\sum_{i=1}^{n}\sum_{j=1}^{m} f(a+\tfrac{2i-1}{2}h,c+\tfrac{2j-1}{2}k), \qquad (6.23)$$
$$h=\frac{b-a}{n},\ k=\frac{d-c}{m}.$$

6.6. Вычисление кратных интегралов. Метод Монте-Карло

Обратимся к вычислению интегралов кратности n. Подходы, основанные на разбиении интервалов интегрирования на m подынтервалов, аналогичные (6.21) и (6.23), требуют объема вычислений порядка m^n и при $n > 3$ (есть приложения, где имеют дело с кратностью 10 и выше) о реальных вычислениях помышлять не приходится. Так, при $n = 10$, $m = 100$ и затратах на вычисление функции в одной точке 1 мкс время вычисления интеграла составит $100^{10} \times 10^{-6}$ с $\approx$ 3 млн. лет.

Сущность метода статистических испытаний (Монте-Карло) [18] состоит в следующем. Для области интегрирования S отыскивается n-мерный параллелепипед $R = [a_i \le x_i \le b_i,\ i = 1, 2, \ldots, n]$ и подвергается «бомбардировке» M случайными, равномерно распределенными в нем точками:

$$X^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\right),\ k = \overline{1, M}\,. \qquad (6.24)$$

Величина

$$J = \frac{1}{N}\sum_{k=1}^{N} f(X^{(k)}), \qquad (6.25)$$

умноженная на объем параллелепипеда $\prod_{i=1}^{n}(b_i - a_i)$, где $N \le M$ − количество точек, попавших в S, может быть принята за оценку интеграла. Кстати, отношение N / M дает соотношение между объемами S и параллелепипеда.

Для отыскания приемлемого значения n можно воспользоваться законом больших чисел и прийти к *оценке, независимой от кратности* интеграла

$$N = \frac{1}{4\varepsilon^2\delta}, \qquad (6.26)$$

где ε − заданная абсолютная погрешность; δ − вероятность ошибки. Если задать $\varepsilon = 0.01$, $\delta = 0.01$, получаем «фантастическое» значение $N = 250\,000$. Умерьте свои требования к точности и надежности оценок, и появится реальная возможность интегрирования для $n > 2$.

Разумеется, ни о какой внекомпьютерной реализации метода Монте-Карло не может быть и речи. В большинстве программных сред имеется *датчик случайных чисел, равномерно распределенных в интервале* (0, 1). Фактически эти числа являются не случайными, а *псевдослучайными* (получаемыми по некоторому алгоритму, но подчиняющимися ряду критериев проверки на случайность). Для формирования случайной точки n-мерного параллелепипеда берем n очередных случайных чисел $0 \le z_i \le 1$ ($i = 1, 2, \ldots, n$) и получаем координаты точки (6.24):

$$x_i^{(k)} = a_i + (b_i - a_i)\, z_i \ (i = 1, 2, \ldots, n).$$

Ниже приведены итоги экспериментов по вычислению значения интеграла $J = \int_0^1 x^2 dx = 0.33333...$ методом Монте-Карло (конечно, так вычислять однократный интеграл − стрелять из пушки по воробьям).

Эксперимент 1	
N	Оценка
10	0.2645
20	0.4672
30	0.3515
40	0.3253
50	0.3492
60	0.3225
70	0.3349
80	0.3865
90	0.3314
100	0.3733
150	0.3388
200	0.3449
250	0.3249
500	0.3192
1000	0.3391
2000	0.3345

Эксперимент 2	
N	Оценка
10	0.3931
100	0.3929
1000	0.3245
10000	0.3319

Эксперимент 3	
N	Оценка
10	0.2814
100	0.2977
1000	0.3296
10000	0.3343
100000	0.3335
1000000	0.3355

Заметим, что методы Монте-Карло давно используют при решении задач, сводимых к поиску экстремумов функций многих переменных. Здесь область поиска «бомбардируется» N случайными точками, среди них отыскивается наилучшая; берется ее окрестность и также подвергается «бомбардировке». Этот процесс продолжается до тех пор, пока область поиска не окажется достаточно малой.

60 лет назад методы Монте-Карло ограничивались вычислением интегралов и задачами диффузии [28]. Сегодня они используются и при моделировании экономических и технологических процессов.

6.7. Численное интегрирование средствами MatLab

MatLab содержит достаточно большой набор функций, предназначенных для интегрирования и решения смежных задач. Так, функция `quad('имя',a,b[,eps])` позволяет вычислить определенный интеграл для интервала (`a`, `b`) с относительной погрешностью `eps` (по умолчанию 10^{-6}) методом Симпсона, `'имя'` – имя подынтегральной функции (встроенной или М-файла). При этом можно пользоваться и так называемыми *анонимными функциями*, которые задаются непосредственно в тексте программы и содержат только один исполняемый оператор. Например, зададим анонимную функцию в виде `F=@(x) x^2`, определяющую значение функции $F(x) = x^2$ при заданном значении аргумента. Тогда оператор `Q=quad('F',0,2)` обеспечивает поиск оценки интеграла от указанной функции в диапазоне от 0 до 2.

Той же цели служат функция `quadl`, базирующаяся на квадратуре Lobatto (по утверждению авторов, лучше чем `quad` в случае гладких функций), и `quadgk`, в основе которой лежит квадратура Гаусса – Кронрода (самая эффективная по точности

для подынтегральных функций, описывающих колебательные процессы).

Для вычисления интегралов кратности 2 и 3 соответственно имеются функции `dblquad('имя',a1,b1,a2,b2[,eps])` и `triplequad('имя',a1,b1,a2,b2,c1,c2[,eps])`.

Иногда полезна функция `polyarea(X,Y)`, предназначенная для вычисления площади многоугольника с координатами вершин (`X,Y`).

Однако будьте готовы и к отказам из-за недостижимости требуемой точности.

Глава 7. ЧИСЛЕННОЕ РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

С обыкновенными дифференциальными уравнениями (*ordinary differential equation*) выпускник средней школы сталкивается при первом же знакомстве с понятием неопределенного интеграла и на предложение решить уравнение $y'(x) = \sin(x)$ бодро сообщает, что $y(x) = \cos(x) + \text{const}$. При изучении азов теоретической механики в школе и вузе он сталкивается с интерпретацией понятий скорости и ускорения как значений первой и второй производных, которые автоматически переносятся и в сферу экономических и социологических исследований – динамика активов некоторого банка, взаимоотношения в природе и обществе и др. Попытаемся напомнить читателю или ознакомить его с простейшими такими уравнениями, постановкой задач и простейшими численными процедурами их решения.

7.1. Постановка и решение задачи Коши

Как мы уже видели выше, решение уравнения $y'(x) = \sin(x)$ представляет собой семейство функций $y(x) = \cos(x) + \text{const}$, т. е. определяется с точностью до произвольной константы, а решением $y''(x) = \sin(x)$ является семейство функций $y(x) = -\sin(x) + c_1 x + c_2$, зависящее уже от двух констант. Для того чтобы выделить в таких семействах конкретную функцию, надо задать соответствующее количество дополнительных условий.

Пусть надо найти функцию $y = y(x)$, являющуюся решением обыкновенного дифференциального уравнения

$$F(x, y, y', y'', \ldots, y^{(n)}) = 0 \qquad (7.1)$$

и удовлетворяющую условиям

$$y(x_0) = p_0,\ y'(x_0) = p_1,\ y''(x_0) = p_2,\ \ldots,\ y^{(n-1)}(x_0) = p_{n-1}. \qquad (7.2)$$

Такая задача называется задачей Коши для уравнения (7.1).

Возьмем простейший вариант (7.1) − линейное уравнение n-го порядка с постоянными коэффициентами

$$a_0 y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \ldots + a_{n-1} y' + a_n y = f(x). \qquad (7.3)$$

Известно, что общее решение уравнения (7.3) складывается из решения соответствующего однородного уравнения

$$a_0 y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \ldots + a_{n-1} y' + a_n y = 0 \qquad (7.4)$$

и частного решения (7.3).

Для решения однородного уравнения возьмем $y = \mathrm{e}^{k x}$ и его подстановкой в (7.4) получаем так называемое *характеристическое уравнение*

$$a_0 k^n + a_1 k^{n-1} + a_2 k^{n-2} + \ldots + a_{n-1} k + a_n = 0. \qquad (7.5)$$

Отыскав его корни, получим искомое решение в виде

$$y(x) = C_1 \mathrm{e}^{k_1 x} + C_2 \mathrm{e}^{k_2 x} + \ldots + C_n \mathrm{e}^{k_n x}. \qquad (7.6)$$

Учитывая то, что среди корней могут быть комплексно-сопряженные $k = a \pm b\,i$, соответствующие слагаемые в (7.6) заменяют на $\mathrm{e}^{a x}(C_1 \cos(b x) + C_2 \sin(b x))$. Если обнаруживается m кратных корней, то соответствующие слагаемые в (7.6) заменяются на $\mathrm{e}^{k x}(C_1 + C_2 x + \ldots + C_{m-1} x^{m-1})$.

Например, для уравнения $y''' - 6y'' + 4y' - 24y = 0$ соответствующее характеристическое уравнение выступает в виде $k^3 - 6k^2 + 4k - 24 = 0$. Его корни равны 6, $2i$ и $-2i$, и общее решение уравнения имеет вид

$$y(x) = C_1 \mathrm{e}^{6x} + C_2 \cos 2x + C_3 \sin 2x\ .$$

Несколько сложнее решается проблема поиска частного (какого-нибудь) решения неоднородного уравнения. Если его правая часть $f(x)$ является алгебраическим многочленом m-й степени, то можно воспользоваться методом неопределенных коэффициентов, представить решение многочленом той же степени, подставить в уравнение и приравнять коэффициенты при соответствующих степенях. Например, для уравнения $y''' - 6y'' + 4y' - 24y = 48x^2 + 8x + 20$ берем $y^*(x) = Ax^2 + Bx + C$ и в итоге получаем систему

$$\begin{aligned} -12A + 4B - 24C &= 20, \\ 8A - 24B &= 8, \\ -24A &= 48. \end{aligned}$$

Откуда находим $A = -2$, $B = -1$, $C = 0$, т. е. общее решение данного уравнения имеет вид

$$y(x) = C_1 \mathrm{e}^{6x} + C_2 \cos 2x + C_3 \sin 2x - 2x^2 - x$$

(семейство решений, определяемое тремя параметрами).

Поставив задачу Коши для приведенного уравнения с условиями при $x_0 = 0$ $y(0) = 1$, $y'(0) = 0$, $y''(0) = 0$, получаем систему

$$y(0) = C_1 + C_2 = 1,$$
$$y'(0) = 6\,C_1 + 2\,C_3 - 1 = 0,$$
$$y''(0) = 36\,C_1 - 4\,C_2 - 4 = 0.$$

Откуда находим $C_1 = 0.2$, $C_2 = 0.8$, $C_3 = -0.1$.

Если правая часть (7.3) − тригонометрический многочлен, то подход к решению аналогичен рассмотренному. В более общем же случае при выборе класса функций, описывающих искомое решение, остается надеяться на интуицию и накопленный опыт решения подобных задач. Например, при столкновении с уравнениями типа

$$\frac{dy}{dx} = x\,y \text{ или } \frac{dy}{dx} = \frac{y}{x^2}$$

возникает мысль о разделении переменных

$$\frac{dy}{y} = x\,dx,\ \frac{dy}{y} = \frac{dx}{x^2}$$

и независимом интегрировании обеих частей уравнений. Откуда получаем

$$\ln y = \frac{x^2}{2} + \text{const} \text{ и } \ln y = -\frac{1}{x} + \text{const}.$$

Если в (7.3) коэффициенты зависят от x или решаемое уравнение нелинейно, за исключением частных случаев, подобных приведенному выше, не следует питать надежд на получение решения в аналитической форме и разумнее сразу обратиться к поиску численного решения.

Заметим, что часто решение задачи для уравнения n-го порядка удобнее свести к решению системы n уравнений первого порядка. Так задача для уравнения (7.3) с начальными условиями (7.2) может быть представлена в виде системы

$$\frac{dy}{dx} = z_1,\ \frac{dz_1}{dx} = z_2,\ \ldots,$$
$$\frac{dz_{n-1}}{dx} = \frac{1}{a_0}\left[f(x) - a_1 z_{n-1} - a_2 z_{n-2} - \ldots - a_{n-1} z_1 - a_n y\right]$$

и $y(x_0) = p_0$, $z_1(x_0) = p_1$, $z_2(x_0) = p_2$, …, $z_{n-1}(x_0) = p_{n-1}$.

7.2. Простейшие методы решения задачи Коши

Сравнительная простота решения рассмотренного уравнения с постоянными коэффициентами и правой частью, зависящей только от x, исчезает в более общем случае. Возьмем уравнение первого порядка $y'(x) = f(x, y)$ с начальным условием $y(x_0) = y_0$ и попытаемся найти $y(x)$ для $x = x_0 + n\,h$, $n = 1, 2, \ldots$ Ограничившись лишь двумя членами разложения в ряд Тейлора, имеем

$$y(x+h) = y(x_n) + h\,y'(x_n) + O(h^2).$$

Введя для краткости записи обозначения $x_n = x_0 + n\,h$, $y_n = y(x_n)$, получаем отсюда

$$y_{n+1} = y_n + h\,f(x_n, y_n),\ n = 0, 1, 2, \ldots. \qquad (7.7)$$

Приведенная формула определяет так называемый *метод Эйлера*, имеющий простую геометрическую интерпретацию (рис. 7.1): из точки (x_n, y_n) проводится касательная к искомой кривой $y = y(x)$ до уровня $x = x_n + h$. Очевидно, что в процессе последовательных переходов по (7.7) погрешность $O(h)$ существенно возрастает. Поэтому метод Эйлера (7.7) в чистом виде применяется лишь для грубых оценок при небольшом количестве точек (узлов).

Чаще используют *модифицированный метод Эйлера*, базирующийся на учете трех членов разложения в ряд Тейлора, имеющий погрешность $O(h^2)$ и использующий промежуточную точку для коррекции положения касательной (рис. 7.2):

$$\begin{aligned} y_{n+\frac{1}{2}} &= y_n + \tfrac{h}{2}\,f(x_n, y_n), \\ y_{n+1} &= y_n + h\,f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}),\quad n = 0, 1, 2, \ldots \end{aligned} \qquad (7.8)$$

Другая модификация метода Эйлера, называемая *модифицированным методом Эйлера – Коши*, с той же погрешностью имеет вид

$$\begin{aligned} y^*_{n+1} &= y_n + h\,f(x_n, y_n), \\ y_{n+1} &= y_n + \tfrac{h}{2}[\,f(x_n, y_n) + f(x_{n+1}, y^*_{n+1})],\quad n = 0, 1, 2, \ldots \end{aligned} \qquad (7.9)$$

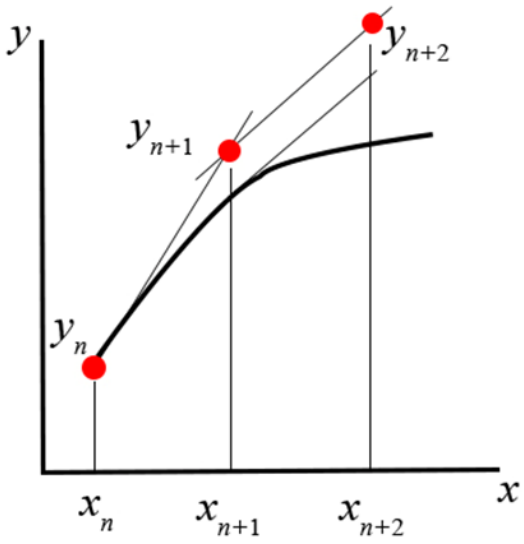


Рис. 7.1

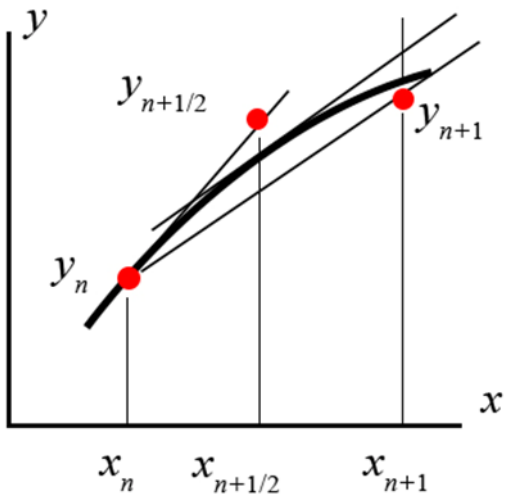


Рис. 7.2

7.3. Методы Рунге – Кутты

Естественно, если учитывать большее количество членов разложения в ряд Тейлора, то можно обеспечить и высокую точность решения (повышенная точность требует и большего объема вычислений). Наиболее популярным является метод Рунге – Кутты* с погрешностью $O(h^4)$, реализуемый в виде

$$
\begin{aligned}
&y_{n+1} = y_n + \tfrac{1}{6}[k_1 + 2\cdot k_2 + 2\cdot k_3 + k_4],\\
&k_1 = h\ f(x_n, y_n),\\
&k_2 = h\ f(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2}),\\
&k_3 = h\ f(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2}),\\
&k_4 = h\ f(x_n + h, y_n + k_3).
\end{aligned}
\qquad (7.10)
$$

Очевидно, что для обеспечения заданной точности при любом методе необходимо выбрать достаточно малый интервал между смежными значениями x_n и x_{n+1}. Однако заранее оценить

---

* Карл Давид Тольме Рунге (1856−1927) − немецкий математик, физик и спектроскопист. Совместно с М. Куттой (1867−1944) разработал методы численного интегрирования систем обыкновенных дифференциальных уравнений. Исследовал поведение полиномиальной интерполяции при повышении степени полиномов (феномен Рунге).

его так, чтобы и точность достигалась, и объем вычислений был разумным, в общем случае нереально. Поэтому прибегают к *системе двойного пересчета*: начинают «прогулку» по интервалу с шагом $h = x_{n+1} - x_n$, затем берут вдвое меньший шаг и проходят интервал в два приема, сравнивают оценки решения при полном и половинном шаге по критерию точности. Если эти оценки достаточно близки, переходят к очередному интервалу и действуют в нем аналогично (можно использовать и полученный уменьшенный шаг). В противном случае уменьшают шаг вдвое, пытаясь получить более точную оценку.

Можно использовать и некоторое уточнение оценки решения y_{n+1} на основании полученных оценок $Y(h)$ и $Y(h/2)$:

$$y_{n+1} = Y(h/2) + \tfrac{1}{15}\left[Y(h/2) - Y(h)\right]. \qquad (7.11)$$

В ряде программных сред часто используется и *метод Кутты – Мерсона*, достоинство которого лишь в том, что в процессе двойного пересчета при выполнении некоторых условий можно и удваивать текущий шаг.

$$\begin{aligned}
y_{n+1} &= y_n + \tfrac{1}{2}[k_1 + 4\,k_4 + k_5],\\
k_1 &= \tfrac{h}{3}\, f(x_n, y_n),\\
k_2 &= \tfrac{h}{3}\, f(x_n + \tfrac{h}{3}, y_n + k_1),\\
k_3 &= \tfrac{h}{3}\, f(x_n + \tfrac{h}{3}, y_n + \tfrac{k_1 + k_2}{2}),\\
k_4 &= \tfrac{h}{3}\, f(x_n + \tfrac{h}{2}, y_n + \tfrac{3}{8}k_1 + \tfrac{9}{8}k_3),\\
k_5 &= \tfrac{h}{3}\, f(x_n + h, y_n + \tfrac{3}{2}k_1 - \tfrac{9}{2}k_3 + 6\,k_4).
\end{aligned} \qquad (7.12)$$

Оценка погрешности определяется величиной

$$D_{n+1} = \tfrac{1}{5}\left|y_{n+1} - W_{n+1}\right|, \qquad (7.13)$$

где

$$W_{n+1} = y_n + \tfrac{3}{2}k_1 - \tfrac{9}{2}k_3 + 6\,k_4 \qquad (7.14)$$

(эта величина вычисляется при поиске k_5).

7.4. Решение задачи Коши для систем уравнений

Рассмотренные методы Эйлера и Рунге распространяются и на системы обыкновенных дифференциальных уравнений.

Пусть отыскивается вектор-функция

$$Y(x) = \{y^{(1)}(x), y^{(2)}(x), \ldots, y^{(m)}(x)\}, \qquad (7.15)$$

являющаяся решением системы уравнений

$$\frac{dy^{(s)}(x)}{dx} = f^{(s)}(x, y^{(1)}(x), y^{(2)}(x), ..., y^{(m)}(x)), s = \overline{1,m}, \qquad (7.16)$$

$$y^{(s)}(x_0) = y_0^{(s)}, s = \overline{1,m}. \qquad (7.17)$$

Если обобщить на эту задачу приведенные ранее формулы метода Рунге-Кутта, то здесь они запишутся в форме

$$y_{n+1}^{(s)} = y_n^{(s)} + \frac{1}{6}[k_1^{(s)} + 2k_2^{(s)} + 2k_3^{(s)} + k_4^{(s)}],$$

$$k_1^{(s)} = hf(x_n, y_n^{(1)}, y_n^{(2)}, ..., y_n^{(m)}),$$

$$k_2^{(s)} = hf(x_n + \frac{h}{2}, y_n^{(1)} + \frac{1}{2}k_1^{(1)}, y_n^{(2)} + \frac{1}{2}k_1^{(2)}, ..., y_n^{(m)} + \frac{1}{2}k_1^{(m)}), \qquad (7.18)$$

$$k_3^{(s)} = hf(x_n + \frac{h}{2}, y_n^{(1)} + \frac{1}{2}k_2^{(1)}, y_n^{(2)} + \frac{1}{2}k_2^{(2)}, ..., y_n^{(m)} + \frac{1}{2}k_2^{(m)}),$$

$$k_4^{(s)} = hf(x_n + h, y_n^{(1)} + k_3^{(1)}, y_n^{(2)} + k_3^{(2)}, ..., y_n^{(m)} + k_3^{(m)}), s = \overline{1,m}.$$

Аналогичные обобщения можно построить и для других рассмотренных выше методов. Незначительные изменения произойдут в системе контроля достижения точности последовательным уменьшением шага. Здесь сравнению с заданной точностью следует подвергать максимальное из значений погрешностей по всем искомым функциям решения.

7.5. Конечноразностные методы и формулы Адамса

Рассмотренные выше методы требуют многократного вычисления значений правой части уравнения. Для уменьшения этой работы можно воспользоваться ее конечноразностной аппроксимацией. Принципиальный подход к построению разностных методов для задачи определяется следующими соображениями.

Пусть отыскивается решение уравнения $y'(x) = f(x, y)$ с начальным условием $y(x_0) = y_0$ для $x = x_0 + n\,h$, $n = 1, 2, ...$ Предположим, что найдены p значений решения y_n для $x_n = x_0 + n\,h$,

$n = 1, 2, \ldots, p$ (методами Эйлера или Рунге). Представим решение уравнения для очередной точки в интегральной форме

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y(x))\, dx,\ n = p, p+1, \ldots \qquad (7.19)$$

и заменим подынтегральную функцию интерполяционным многочленом, совпадающим в $p + 1$ точке со значениями $f(x_k, y_k) = f_k$, $k = 0, 1, \ldots, p$. Если в качестве такого полинома взять формулу Ньютона интерполирования вперед

$$f(x, y(x)) \equiv Q_p(x) = f_n + t\,\Delta f_n + \frac{t(t+1)}{2!}\Delta^2 f_n + \ldots + \frac{t(t+1)\ldots(t+p-1)}{p!}\Delta^{(p)} f_n,$$

$$t = (x - x_0) / h$$

и подставить в (7.19), получим представление решения в очередной точке через конечные разности, определяемые оценками $f(x, y)$ в предыдущих $p + 1$ точках:

$$y_{n+1} = y_n + h\sum_{k=0}^{p} \alpha_k \Delta^{(k)} f_n,\ n \geq p, \qquad (7.20)$$

где $\alpha_0 = 1,\ \alpha_k = \frac{1}{k!}\int_0^1 t(t+1)\ldots(t+k-1)\, dt,\ k = \overline{1, p}$ (погрешность такой аппроксимации имеет порядок $O(h^{p+2})$). Формулы (7.20) называются *экстраполяционными формулами Адамса*.

Так при $p = 3$ получим *экстраполяционную формулу Адамса*

$$y_{n+1} = y_n + h\,[\, f_n + \tfrac{1}{2}\Delta f_n + \tfrac{5}{12}\Delta^2 f_n + \tfrac{3}{8}\Delta^3 f_n\,],\ n \geq 3 \qquad (7.21)$$

или в развернутом виде

$$y_{n+1} = y_n + \frac{h}{24}\,[55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}],\ n \geq 3. \qquad (7.22)$$

Последовательное применение этой формулы при известных значениях $y(x)$ в точках x_0, x_1, x_2, x_3 позволяет найти решение задачи для всех последующих точек.

Базируясь на формуле Ньютона интерполирования назад с узлами x_k, $k = 1, 2, \ldots, p + 1$

$$Q_p(x) = f_{n+1} + (t-1)\Delta f_{n+1} + \frac{(t-1)t}{2!}\Delta^2 f_{n+1} + \ldots + \frac{(t-1)t\ldots(t+p-2)}{p!}\Delta^{(p)} f_{n+1},$$

$$t = (x - x_n) / h,$$

аналогично для $p = 3$ получим *интерполяционную формулу Адамса*

$$y_{n+1} = y_n + \frac{h}{24}\left[9 f_{n+1} + 19 f_n - 5 f_{n-1} + f_{n-2}\right], n \geq 3. \qquad (7.23)$$

7.6. Разностные методы для краевых задач. Метод прогонки

Пусть требуется найти функцию $y = y(x)$, являющуюся решением обыкновенного дифференциального уравнения

$$F(x, y, y', y'', \ldots, y^{(n)}) = 0$$

и удовлетворяющую n условиям на функцию и ее производные, заданным *на концах отрезка* $[A, B]$. В такой постановке мы имеем дело с так называемой *краевой задачей* для обыкновенного дифференциального уравнения.

Возьмем для примера линейное уравнение второго порядка

$$a(x)\frac{d^2y}{dx^2} + b(x)\frac{dy}{dx} + c(x)\, y = f(x) \qquad (7.24)$$

и рассмотрим различные формы краевых условий.

Краевые условия типа

$$y(A) = \varphi_1(A),\ y(B) = \varphi_2(B) \qquad (7.25)$$

определяют *первую краевую задачу*, или *задачу Дирихле*.

Краевые условия с заданием значений производных

$$y'(A) = \varphi_1(A),\ y'(B) = \varphi_2(B) \qquad (7.26)$$

определяют *вторую краевую задачу*, или *задачу Неймана*.

Существует и *третья краевая задача* (*со смешанными условиями*) с краевыми условиями

$$\alpha_1 \frac{dy(A)}{dx} + \beta_1\, y(A) = \gamma_1,\ \alpha_2 \frac{dy(B)}{dx} + \beta_2\, y(B) = \gamma_2. \qquad (7.27)$$

Такого рода задачи, в большинстве своем, связаны с решением физических проблем.

Различным методам решения краевых задач посвящена обширная литература. Здесь мы рассмотрим самые простейшие из разностных методов, не вдаваясь в подробности. Основной подход к решению такого рода задач связан с использованием аппарата конечноразностной аппроксимации, когда область изменения аргумента заменяется сеткой узлов $x_k = x_0 + k\,h$,

$k = 0, 1, \ldots, n$ и производные заменяются их конечноразностным представлением с той или иной точностью аппроксимации. Так при точности порядка $O(h^2)$ можно использовать представления

$$\frac{dy_k}{dx} = \frac{y_{k+1} - y_{k-1}}{2h} + O(h^2) \ , \ k = \overline{1, n-1} \ ;$$

$$\frac{d^2 y_k}{dx^2} = \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + O(h^2) \ , \ k = \overline{1, n-1} \ ; \tag{7.28}$$

$$\frac{dy_0}{dx} = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2); \tag{7.29}$$

$$\frac{dy_n}{dx} = \frac{y_{n-2} - 4y_{n-1} + 3y_n}{2h} + O(h^2). \tag{7.30}$$

Разумеется, можно использовать и более точные аппроксимации с большим числом учитываемых узлов.

Если рассмотреть уравнение (7.24)

$$a(x)\frac{d^2 y}{dx^2} + b(x)\frac{dy}{dx} + c(x)\, y = f(x), \tag{7.31}$$

его аппроксимация во внутренних узлах сетки дает систему $n - 1$ линейных алгебраических уравнений с $n + 1$ неизвестными

$$a(x_k)\frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + b(x_k)\frac{y_{k+1} - y_{k-1}}{2h} + c(x_k)y_k = f(x_k). \tag{7.32}$$

Если выбранная система хорошо аппроксимирует исходное дифференциальное уравнение и ее решение непрерывно зависит от изменения правой части уравнения и начальных (граничных) условий (в этом случае говорят об устойчивости схемы), то при $h \to 0$ ее решение сходится к решению поставленной задачи [2].

Вводя обозначения

$$A_k = a(x_k) - \frac{h}{2}b(x_k), B_k = -2 \ a(x_k) + h^2 \ c(x_k),$$
$$C_k = a(x_k) + \frac{h}{2}b(x_k), D_k = h^2 \ f(x_k), \tag{7.33}$$

получаем систему линейных алгебраических уравнений

$$A_k \ y_{k-1} + B_k \ y_k + C_k \ y_{k+1} = D_k, k = \overline{1, n-1}. \tag{7.34}$$

Допустим, что для рассматриваемого уравнения поставлена задача Дирихле, т. е. значения y_0 и y_n − заданные величины. Решение (7.34) методом Гаусса при больших n (реальные физические задачи – источники такого рода уравнений – требуют n порядка сотен) нереально из-за колоссальной вычислительной погрешности, итерационный путь требует значительных временных затрат. С учетом трехдиагональности матрицы коэффициентов можно воспользоваться уже ранее рассмотренным специальным приемом − *методом прогонки* (см. п°2.4). На этом пути к тому же объем хранения матрицы коэффициентов n^2 можно уменьшить до $3n - 2$.

Пусть

$$y_k = P_k y_{k+1} + Q_k,\ k = 0, 1, \ldots, n - 1. \qquad (7.35)$$

Подставляя в (7.34), получаем

$$A_k\ (P_{k-1}\ y_k + Q_{k-1}) + B_k\ y_k + C_k\ y_{k+1} = D_k,\ k = \overline{1, n-1}. \qquad (7.36)$$

Откуда находим

$$P_k = -\frac{C_k}{B_k + A_k\ P_{k-1}}\ ,\ Q_k = \frac{D_k - A_k\ Q_{k-1}}{B_k + A_k\ P_{k-1}}\ ,\ k = \overline{1, n-1}. \qquad (7.37)$$

Поскольку величина y_0 известна, то можно принять $P_0 = 0$, $Q_0 = y_0$ и в соответствии с (7.36) найти прогоночные коэффициенты. Зная величину y_n, можно с помощью (7.35) обратным ходом (в порядке убывания k) найти искомое решение.

7.7. Коротко об уравнениях в частных производных

До какой температуры разогревать моторы самолета на земле, чтобы они не «зачихали» на взлете? Какой должна быть защита ядерного реактора, чтобы при землетрясении в 8 баллов не возникала утечка радиации? Какой должна быть начальная скорость для вывода ракеты на орбиту при полетах к Марсу? Какова должна быть температура горячей воды на выходе из источника теплоснабжения, чтобы жильцы многоэтажного дома не спали в «заячьих тулупчиках» и не задыхались от 30-градусной жары? Как повлияет ураган на Багамах на метеорологическую обстановку в Мурманске?

Ушло время, когда Мария Склодовская-Кюри и Анри Беккерель носили коробочку с радием в собственных карманах. Ушло

время, которое знаменитый физик Р. Фейнман назвал «дерганием дракона за хвост», когда Луи Слотин в Лос-Аламосе опытным путем определял критическую массу ядер плутония для атомной бомбы и погиб от лучевой болезни. Уже не строят плотины, опираясь лишь на опыт бобров, и отели в Нью-Васюках по проектам отелей Майами. Прогноз погоды в Сибири сегодня узнают в интернете, а не на основе народных примет Новгородской области.

На смену рискованному физическому (подчас и социологическому) эксперименту пришел объективный расчет, базирующийся на физико-математических моделях и применении так называемых уравнений математической физики. С помощью этих уравнений описывают процессы переноса нейтронов, обтекания крыла летательного аппарата, распространения взрывной волны, течения жидкостей и газов, распределения температуры в многослойных средах (космических скафандрах, оболочках ракет, стенах панельных домов), распределения напряжений для шахтной крепи и т. д. Уравнения в частных производных, часто называемые уравнениями математической физики, базируются на функциях не одной, а нескольких переменных.

Среди них особо выделяют линейные уравнения второго порядка, приводимые *к канонической форме*:

$$\sum_{i=1}^{n} \lambda_i(X)\frac{\partial^2 Z(X)}{\partial x_i^2} + \sum_{i=1}^{n} b_i(X)\frac{\partial Z(X)}{\partial x_i} + c(X)\,Z(X) + f(X) = 0. \quad (7.38)$$

Все такие уравнения условно разделяют на три типа:

1) если число положительных значений λ_i (или отрицательных) равно n, уравнение называют *эллиптическим*;

2) если все λ_i отличны от нуля и лишь одно из них положительно (или отрицательно), уравнение называют *гиперболическим* (если все λ_i отличны от нуля, но различны по знаку, уравнение называют *ультрагиперболическим*);

3) если хотя бы один коэффициент λ_i равен нулю, а остальные имеют одинаковые знаки, уравнение называют *параболическим*.

Тип уравнения может меняться в зависимости от точки – места или времени действия изучаемой системы. Так *уравнение Трикоми*

$$y\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}=0 \qquad (7.39)$$

является эллиптическим при $y > 0$ и гиперболическим при $y < 0$.

Простейшим примером уравнения эллиптического типа является *уравнение Пуассона*

$$\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}+ f(x_1, x_2)=0, \qquad (7.40)$$

которое при $f(x_1, x_2) = 0$ называется *уравнением Лапласа*. Для него обычно ставятся граничные условия − задача Дирихле, где задано поведение искомой функции на границе области его действия. Обычно эллиптические уравнения используются при описании стационарных, не зависящих от времени процессов распределения тепла, колебаний, течения несжимаемой жидкости, рассеяния (дифракции) и т. д.

Примерами гиперболических уравнений могут служить уравнения, возникающие при описании малых колебаний струн и мембран, акустических и электромагнитных колебаний:

волновое уравнение

$$\frac{\partial^2 U}{\partial t^2}=\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}, \qquad (7.41)$$

телеграфное уравнение

$$\frac{\partial^2 U}{\partial t^2}-c^2\frac{\partial^2 U}{\partial x^2}+(\alpha+\beta)\frac{\partial U}{\partial t}+\alpha\beta U=0, \qquad (7.42)$$

а также ряд уравнений механики сплошных сред (в частности, уравнения течения жидкости и газовой динамики). Задачи, поставленные для этих уравнений, предполагают задание не только граничных, но и начальных условий.

Типичным примером параболического уравнения служит одномерное *уравнение теплопроводности* − частный случай уравнения диффузии, используемого при описании процессов диффузии частиц и распространения тепла в некоторой среде:

$$\frac{\partial}{\partial x}\left(k\frac{\partial U}{\partial x}\right)+F(x,t)=c\ \rho\ \frac{\partial U}{\partial t}. \qquad (7.43)$$

Здесь рассматривается тонкий теплоизолированный стержень длины l, на концах которого с некоторого момента включены

тепловые источники и возникает тепловой поток от более нагретого конца к менее нагретому. Через $U(x, t)$ обозначена температура стержня в сечении x в момент t, c – удельная теплоемкость, ρ – плотность материала, $F(x, t)$ − плотность источников тепла в самом стержне и $k(x)$ − коэффициент теплопроводности. При отсутствии внутренних источников (7.43) приведется к виду

$$\frac{\partial U}{\partial t} = a^2 \frac{\partial^2 U}{\partial x^2} + f(x,t). \qquad (7.44)$$

Аналогично возникают двух- и трехмерные задачи для тел различной конфигурации (цилиндры, конусы, шары и др.). Задачи, поставленные для этих уравнений, предполагают задание как граничных, так и начальных условий.

За три столетия активной работы физиков и математиков построено множество таких уравнений и аналитических методов их решения. Увы, они применимы только для идеальных объектов и посильны для специалистов с незаурядной математической подготовкой. С появлением ЭВМ ушло в прошлое решение задач не только вручную, но и с помощью аналоговых устройств. Труд аналогового интегратора в течение двух суток сменился на двадцатиминутную работу популярной в шестидесятые годы отечественной ЭВМ М-20.

Численное решение краевых задач, в частности, для уравнений эллиптического и параболического типов базируется на использовании разностных схем. Например, попробуем решить уравнение

$$\frac{\partial U}{\partial t} = a^2 \frac{\partial^2 U}{\partial x^2} + f(x,t)$$

с начальными условиями $U(x,t)_{|t=t_0} = \varphi(x), x \in [0, l]$ и краевыми условиями I рода $U(0, t) = \psi_1(t)$, $U(l, t) = \psi_2(t)$, $t \in [0, T]$. Покроем область решения $[0 \le x \le l, t_0 \le t \le T]$ сеткой с шагом h по x и шагом τ по t. Заменяя производные их разностными аналогами, строим разностную аппроксимацию уравнения

$$\frac{U_i^{k+1} - U_i^k}{\tau} = \left\{ a^2 \frac{U_{i+1}^k - 2\ U_i^k + U_{i-1}^k}{h^2} \right\} + f_i^{k+\frac{1}{2}}, \qquad (7.45)$$

где $i = 1, 2, \ldots, N-1$, $k = 0, 1, \ldots$ Из начальных и граничных условий имеем

$U_i^0 = \varphi(x_i)$, $i = 1, 2, \ldots, N-1$; $U_0^k = \psi_1(t_k)$, $U_N^k = \psi_2(t_k)$, $k = 0, 1, \ldots$

Соответственно идея решения: поскольку начальный по t «слой» $U(x, 0) \equiv U_i^0$ известен, равно как и граничные значения U_0^k, U_N^k при всех t, то ищем значения U_1^k, U_2^k и т. д.

Поскольку такая *явная* схема условно устойчива, предлагается *неявная* схема

$$\frac{U_i^{k+1} - U_i^k}{\tau} = \left\{ a^2 \frac{U_{i+1}^{k+1} - 2U_i^{k+1} + U_{i-1}^{k+1}}{h^2} \right\} + f_i^{k+\frac{1}{2}}, \qquad (7.46)$$

сводящая решение задачи к последовательному решению систем линейных алгебраических уравнений с трехдиагональной матрицей коэффициентов, для которых и создавался метод прогонки.

Переход от одномерной задачи к двумерной, в случае неявных разностных схем, существенно усложняет решение задач и вызывет разработку новых методов. Так, решая уравнение теплопроводности

$$\frac{\partial U}{\partial t} = a_1^2 \frac{\partial^2 U}{\partial x_1^2} + a_2^2 \frac{\partial^2 U}{\partial x_2^2} + f(t, x_1, x_2) \qquad (7.47)$$

с заданными начальными и граничными условиями для прямоугольной области, по аналогии с одномерным случаем покрываем указанную область сеткой с шагами h_1 и h_2 по пространственным координатам и выбираем явную разностную схему с порядком аппроксимации $O(\tau)$. Но эта схема требует выполнения жестких условий сходимости (при h_1 и h_2 порядка 0.01 шаг по t имеет порядок 0.0001) и солидного времени вычислений.

Неявная разностная схема приводит к необходимости решать систему $N_1 \times N_2$ алгебраических уравнений (N_1 и N_2 – число узлов сетки по обеим координатам). При N_1 и N_2 порядка 100 решаем 10000 уравнений! Во избежание такого фантастического расчета выдвинута идея расщепления (продольно-поперечной прогонки) и соответствующий метод дробных шагов [20]. Как вариант, схема последовательной прогонки по одной и затем по другой координатам с точностью аппроксимации $O(\tau^2)$

$$\frac{U_{ij}^{k+0.5}-U_{ij}^{k}}{\tau}=a_1^2\frac{U_{i-1j}^{k+0.5}-2U_{ij}^{k+0.5}+U_{i+1j}^{k+0.5}}{h_1^2}+\frac{1}{2}f_{ij}^{k},\ i=\overline{1,N_1-1},\ j=\overline{1,N_2-1};$$

$$\frac{U_{ij}^{k+1}-U_{ij}^{k+0.5}}{\tau}=a_2^2\frac{U_{ij-1}^{k+1}-2U_{ij}^{k+1}+U_{ij+1}^{k+1}}{h_2^2}+\frac{1}{2}f_{ij}^{k+1},\ j=\overline{1,N_{21}-1},i=\overline{1,N_1-1}.$$

На этом пути решаем 200 систем, но 100-го порядка с трехдиагональными матрицами коэффициентов (сопоставьте объем работ!).

Мы не затронули и тысячной доли полезной информации об уравнениях в частных производных. Остается лишь заметить, что объемы вычислительной работы при решении реальных задач настолько велики даже для современных компьютеров, что это привело к идее параллельных вычислений и созданию суперкомпьютеров [45, 46].

7.8. Обыкновенные дифференциальные уравнения и MatLab

В системе MatLab предусмотрена группа функций, позволяющих решить задачу Коши для систем обыкновенных дифференциальных уравнений, заданных как в явной форме $\frac{dx}{dt}=F(t,x)$, так и в неявной $M(t,x)\frac{dx}{dt}=F(t,x)$ – так называемые *решатели ОДУ* (`solver ODE`). Они предоставляют пользователю возможность выбора метода, задания начальных условий и др.

В простейшем варианте достаточно воспользоваться командой `[T, X]=solver('fun', [t0, tk], X0)`, где значения `t0` и `tk` определяют диапазон интегрирования, `X0` – вектор начальных значений, `fun` – имя функции вычисления правых частей системы, `solver` – имя используемой функции (`ode45` – метод Рунге – Кутты 4- и 5-го порядков, `ode23` – тот же метод 2- и 3-го порядков, `ode113` – метод Адамса для так называемых нежестких систем, `ode23s`, `ode15s` – для жестких систем и др.). Здесь под жесткостью понимается повышенное требование к точности – использование минимального шага во всей области интегрирования.

Версии решателя различаются используемыми методами (по умолчанию относительная погрешность 10^{-3} и абсолютная 10^{-6}),

соответственно временем и успешностью решения. При задании диапазона в виде `[t0, tk]` число элементов в выходных массивах `T` и `X` определяется необходимым для обеспечения точности шагом; при задании его в виде `[t0, t1, t2, ..., tk]` или `[t0:Δt:tk]` – указанными значениями.

Так в простейшем варианте решения уравнения $\frac{dx}{dt} = t\,\mathrm{e}^{-t}$ в интервале $t \in [0, 0.5]$ с начальным условием $x(t = 0) = 1$ достаточно создать функцию с любым именем (например, `function f=texp(t, x)/f=t*exp(-t))` и командой `[T, X]=ode23('texp', [0, 0.5], 1)` получить массивы значений аргумента `T` и соответствующих решений `X`.

При решении задачи $\frac{d^2x}{dt^2} = x + 2\,\mathrm{e}^t, x(0) = 0, \frac{dx}{dt}(t = 0) = 1,$ сводимой к решению системы $\frac{dx_1}{dt} = x_2; \frac{dx_2}{dt} = x_1 + 2\,\mathrm{e}^t$ с начальными условиями $x_1(0) = 0$, $x_2(0) = 1$ при $t \in [0, 2]$ задаем

```
function f=odu2(t, x)/f=zeros(2, 1);
 f(1)=x(2); / f(2)=x(1)+2*exp(t);
```

и оператором `[T,X]=ode15s('odu2',[0:0.5:2],[0 1])` выводим массив аргументов `T` и массив решений `X=[x1, x2≡x]`.

Библиотека функций MatLab в сочетании с превосходной двумерной и даже трехмерной графикой позволяет решать задачи из разных сфер исследований. Примером подобных задач служит известная *задача динамики популяций* − модель взаимодействия «жертв» и «хищников», в которой учитывается уменьшение численности представителей одной стороны с ростом численности другой. Модель была создана для биологических систем, но с определенными корректурами применима к конкуренции фирм, строительству финансовых пирамид, росту народонаселения, распространению эпидемий, перенаселенности, ядерной энергетике, обширной экологической проблематике и др. [39, 47].

Так известная модель Лотки – Вольтерры системы «хищник – жертва» с логистической поправкой описывается системой уравнений

$$\frac{dx_1}{dt} = (a - bx_2)x_1 - \alpha x_1^2,$$

$$\frac{dx_2}{dt} = (-c + dx_1)x_2 - \alpha x_2^2$$

при заданной численности «жертв» и «хищников» в начальный момент $t = 0$. Эта система уравнений относится к числу так называемых *автономных* (или *динамических*), где переменная t в правую часть системы явно не входит. Соответственно можно не только найти решения $x_1 = x_1(t)$, $x_2 = x_2(t)$, но и отобразить связь между ними. В параметрическом задании линия $x_1 = x_1(t)$, $x_2 = x_2(t)$ определяет *фазовую кривую* (*траекторию*) системы − гладкую кривую без самопересечений, замкнутую кривую или точку, позволяя судить об устойчивости системы.

Нижеприведенная программа при задании различных значений α создает соответствующие фазовые портреты (рис. 7.3, 7.4) − обычный колебательный процесс и постепенная гибель популяций.

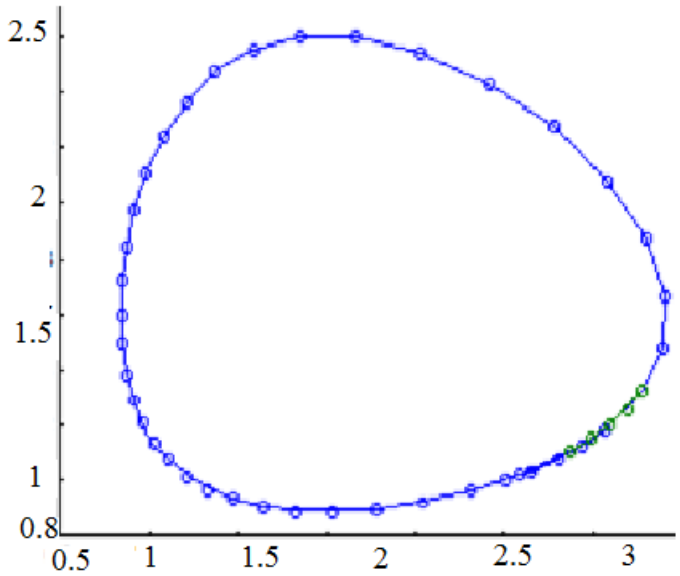


Рис. 7.3

```
function f=VolterraLog(t, x)
a=4; b=2.5; c=2; d=1; alpha=0.1;
f(1)=(a-b*x(2))*x(1)-alpha*x(1)^2;
f(2)=(-c+d*x(1))*x(2)-alpha*x(2)^2; f=f';
» opt=odeset('OutputSel', [1 2], 'OutputFcn',
'odephas2');
» [T, X]=ode45('VolterraLog', [0 10], [3 1],
opt);
```

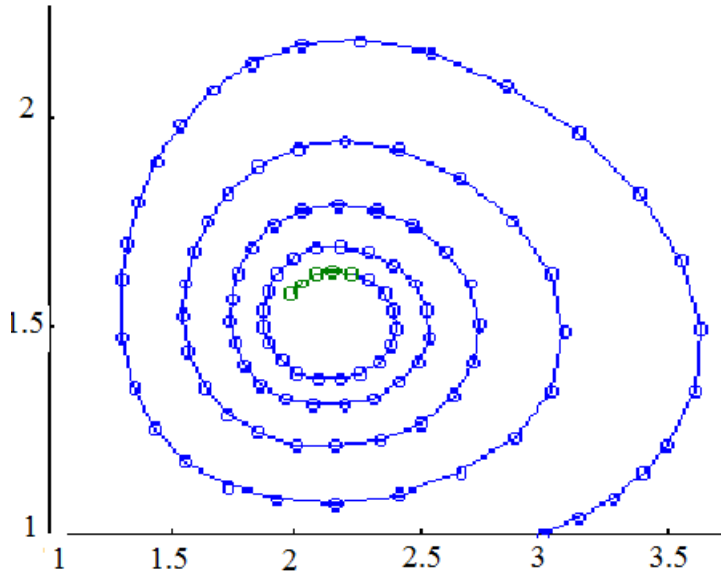


Рис. 7.4

Имеется возможность построения и трехмерного фазового портрета с помощью функции `odephas3`. Например, решение трехмерной задачи Эйлера свободного движения твердого тела, поставленной в виде

$$\frac{dx_1}{dt} = x_2 x_3,\ \frac{dx_2}{dt} = -x_1 x_3,\ \frac{dx_3}{dt} = -0.51\, x_1 x_2,$$

$$x_1(0) = x_2(0) = x_3(0) = 0,$$

выступает в форме программы с описанием функций правой части

```
function f=Eiler(t, x)
f(1)=x(2)*x(3); f(2)=-x(1)*x(3);
f(3)=-0.51*x(1)*x(2); f=f';
```

и операторов решения задачи (первое обращение к `ode45` обеспечивает расчет с автоматическим выбором шага

и построением фазового портрета, второе дает визуализацию найденных трех решений при фиксированном выборе шага).

```
» opt=odeset('OutputSel', [1 2 3], 'OutputFcn',
'odephas3');
» [T, X]=ode45('Eiler', [0 7.25], [0 0 1], opt);
» plot(T, X) % рис. 7.5
» [T, X]=ode45('Eiler', [0:0.25:7.25], [0 1 1]);
» plot(T, X) % рис. 7.6
```

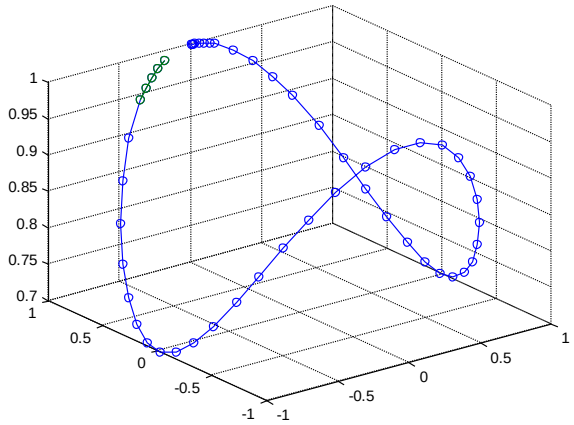


Рис. 7.5

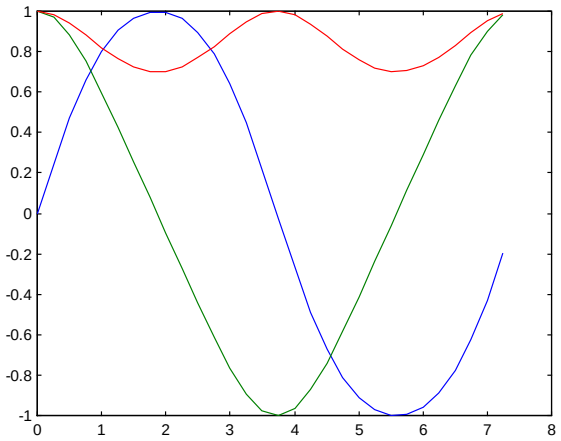


Рис. 7.6

Глава 8. МЕТОДЫ ОПТИМИЗАЦИИ

8.1. Одномерная оптимизация

В случае одной переменной поиск экстремума функции при наличии ограничений или отсутствии таковых не вызывает затруднений. Так для поиска максимума функции $F(x)$ на отрезке $[a, b]$ берут производную, решают уравнение $F'(x) = 0$ (выше мы рассмотрели многообразие соответствующих методов), вычисляют $F(x)$ в найденных «критических» точках, лежащих внутри отрезка и на его концах, с последующим выбором максимального значения из полученных.

Если нет желания искать все корни уравнения $F'(x) = 0$, то для «достаточно гладких» функций можно разбить $[a, b]$ на N частей (10, 100 и т. п.) длиной $h = (b - a) / N$, выделить точку x^* с наибольшим значением $F(x)$, взять ее окрестность $(x^* \pm h)$, разбить на $2N$ частей и повторить процедуру. Эти действия повторять до тех пор, пока длина очередного интервала не станет сопоставимой с заданной точностью.

Сегодня, работая в системах с нормальной графикой и не желая размышлять, задают таблицу значений x, находят массив соответствующих значений $F(x)$, выводят на дисплей в виде графика (в MatLab это позволяет сделать команда `plot(X, F)`), визуально оценивают район максимума (максимумов) и проделывают ту же работу с другими интервалами.

Находятся «оригиналы», которые при требуемой точности 10^{-6} разбивают $[a, b]$ на 10^6 частей и табулируют $F(x)$ с последующим выбором экстремальной точки.

Если эти пути не устраивают из-за солидного объема вычислений, то можно прибегнуть к идее *градиентного спуска* (здесь она не заслуживает столь мудрого названия). Выбираем точку x_0, находим значения $F(x_0)$ и $F'(x_0)$ и смещаемся от этой точки влево или вправо в зависимости от знака $F'(x_0)$ на выбранный шаг h. Если в новой точке значение $F(x)$ меньше значения в предыдущей, уменьшаем шаг вдвое (втрое?, в 10 раз?), возвращаемся

в предыдущую точку и повторяем попытку перехода. В противном случае продолжаем процесс переходов либо до выхода на концы исходного интервала, либо до сведения шага к приемлемой малости. На этом пути есть опасность выйти на локальный, а не глобальный максимум (все зависит от x_0 и начального шага).

В случае унимодальных функций (имеющих единственный экстремум на интервале) можно обойтись и существенно меньшими трудозатратами, если прибегнуть к *методу Фибоначчи*, родственному ему *методу золотого сечения* или к методам квадратичной или кубической интерполяции (*методам Пауэлла* и *Давидона*).

8.1.1. Экстремум унимодальной функции и метод Фибоначчи

Последовательность натуральных чисел, определяемая в форме

$$F_0 = F_1 = 1,\ F_k = F_{k-1} + F_{k-2},\ k \geq 2,$$

называется числами Фибоначчи* (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, …). Обратите внимание на то, что уже 20-е число Фибоначчи превышает 10^4, а 25-е превышает 10^5 и т. д.

Можно доказать, что для функции $y = f(x)$, унимодальной на отрезке $[0, F_n]$, точка максимума может быть локализована в интервале единичной длины путем вычисления и взаимного сравнения не более n значений $f(x)$.

Если обратиться к практике вычислений, исходный интервал можно уменьшить в 100 000 раз в результате лишь 25 вычислений. В самом деле, берем отрезок $[a, b]$ и с учетом заданной точности ε

---

* Леонардо Пизанский (1170−1250), известный под прозвищем Фибоначчи, − первый крупный математик средневековой Европы. Принес в Европу десятичную систему счисления и арабские цифры, в своей «Книге абака» изложил почти все математические сведения того времени, в том числе понятия о пропорциях, обыкновенных дробях, прогрессиях, числах Фибоначчи, способах приближенного извлечения квадратного и кубического корней, квадратных уравнениях, отрицательных числах и др.

найдем номер n числа Фибоначчи такой, что $F_n > (b - a) / \varepsilon$. Возьмем в интервале точку, отстоящую от его начала на F_{n-1} / F_n доли его длины, т. е.

$$x^* = a + (b - a)\frac{F_{n-1}}{F_n},$$

и вычислим значение $f^* = f(x^*)$. Возьмем точку $x = a + b - x^*$ (в сущности, точку с отступом от a на долю F_{n-2} / F_n длины интервала). Если $f(x) < f^*$, точка максимума при $x < x^*$ лежит в интервале $[x, b]$ (рис. 8.1 – заменяем текущее значение a на x) и при $x > x^*$ – в интервале $[a, x]$ (заменяем b на x). В случае $f(x) > f^*$ при $x < x^*$ берем интервал $[a, x^*]$, а при $x > x^*$ – интервал $[x^*, b]$, запоминая $f(x)$ в качестве f^*. В любом случае полученный интервал сопоставляется значению F_{n-1} и новая точка x^* выбирается с учетом соотношения F_{n-2} / F_{n-1}. Эту процедуру повторяем n раз, что гарантирует достижение заданной точности.

Рис. 8.1

Так для функции $f(x) = x\,(1 - x)$ на отрезке [0, 1] поиск максимума с точностью 0.0001 требует 20 вычислений значений функции ($F_{20} = 10946$).

k	a	b	x^*	$f(x^*)$	x	$f(x)$
1	0	1	0.61803	0.23607	0.38197	0.23607
2	0	0.61803	0.38197	0.23607	0.23067	0.18034
3	0.23607	0.61803			0.47214	0.24922
4	0.38197	0.61803	0.47214	0.24922	0.52786	0.24922
5	0.47214	0.61803	0.52786	0.24922	0.56231	0.24612
6	0.47214	0.56231			0.50658	0.24996
7	0.47214	0.52768	0.50658	0.24996	0.49342	0.24996
8	0.47214	0.50658	0.49342	0.24996	0.48529	0.24978
9	0.48529	0.50658			0.49845	0.25000
10	0.49342	0.50658	0.49845	0.25000	0.50155	0.25000
...	....	....	....	....	....	....
20	0.49991	0.50018	0.50009	0.25000	0.50000	0.25000

8.1.2. Экстремум унимодальной функции и золотое сечение

Берем соотношение для чисел Фибоначчи $F_k = F_{k-1} + F_{k-2}$ и ищем F_k в виде r^k. Прямой подстановкой получаем $r^k = r^{k-1} + r^{k-2}$. Откуда видно, что r является корнем уравнения $r^2 = r + 1$ и соответственно

$$F_k = C_1\left(\frac{1+\sqrt{5}}{2}\right)^k + C_2\left(\frac{1-\sqrt{5}}{2}\right)^k,$$

где C_1 и C_2 − произвольные константы, которые можно найти из условия $F_0 = F_1 = 1$,

$$F_0 = C_1 + C_2 = 1,\ F_1 = C_1\left(\frac{1+\sqrt{5}}{2}\right) + C_2\left(\frac{1-\sqrt{5}}{2}\right)$$

в виде

$$C_1 = 1 + \frac{1}{\sqrt{5}},\ C_2 = 1 - \frac{1}{\sqrt{5}}.$$

При больших значениях k с достаточно высокой точностью

$$F_k = \left(1 + \frac{1}{\sqrt{5}}\right)\left(\frac{1+\sqrt{5}}{2}\right)^k$$

и отношение $\frac{F_k}{F_{k-1}} = \frac{1+\sqrt{5}}{2} \approx 1.62$. Это число называют *золотым сечением* (в Древней Греции считалось, что прямоугольник с отношением сторон $1 / 1.62 \approx 0.62$ имеет самые приятные пропорции, что нашло отражение в архитектуре). Оно позволяет построить видоизменение рассмотренного выше метода. Здесь без предварительного поиска чисел Фибоначчи выбираются начальные точки на удалении $0.62\,(b - a)$ от концов промежутка и ранее описанный процесс продолжается до тех пор, пока интервал поиска не окажется меньше допустимой погрешности.

8.2. Многомерная оптимизация без учета ограничений

В случае функции n переменных традиционный классический путь взятия частных производных и решения соответствующей системы (как правило, нелинейных) уравнений при отсутствии информации о «районе дислокации» корня упирается в выбор начального приближения, гарантию сходимости итерационного

процесса, в наличие множества корней. При появлении ограничений добавляется и проблема оценки значений функции на множестве точек границы.

Можно пойти другим путем, взяв область «прямоугольной» формы, по каждой координате выбрать 100 точек и вычислить значения функции на образовавшейся сетке из 100^n точек. В этой ситуации уже при $n > 3$ ни о каких реальных расчетах говорить не приходится.

Методы поиска экстремума функции без учета ограничений в принципе можно разделить на методы прямого поиска и градиентные методы.

8.2.1. Методы прямого поиска

Если есть уверенность в том, что функция унимодальна (имеет единственный минимум / максимум), напрашивается простой *метод покоординатного спуска* (*подъема*).

Пусть стоит задача максимизации. Берем некоторую точку, выясняем направление возрастания функции по x_1 (либо оценкой значения частной производной функции по x_1, либо вычислением функции в смежных точках, отстоящих от выбранной «влево» и «вправо» на заданный шаг h) и идем в выбранном направлении с этим шагом, пока значения функции возрастают. Затем для полученной точки выясняем направление роста функции по x_2 и проводим аналогичную прогулку по этой переменной. Для остальных переменных проделываем аналогичную процедуру (шаг переходов лучше для каждой из переменных выбирать свой). Далее уменьшаем шаг (шаги) в несколько (2, 3 или 10?) раз и повторяем описанный выше процесс до тех пор, пока величина шага (шагов) не окажется в пределах выбранной точности.

Например, при поиске максимума функции

$$f(x_1, x_2) = 100 - x_1^2 - (x_2 - 3)^2$$

берем за начальную точку (10, 10) и шаг $h = 2$ по обеим переменным. Имеем $f(10, 10) = -49$, и поскольку $f(12, 10) = -70 < f(10, 10)$, идем в другом направлении по x_1, получая $f(8, 10) = -13$;

$f(6, 10) = 15$; $f(4, 10) = 35$; $f(2, 10) = 47$; $f(0, 10) = 51$; $f(-2, 10) = 47 < f(0, 10)$.

Принимаем за отправную точку (0, 10) и аналогично идем по переменной x_2, получая $f(0, 8) = 75$; $f(0, 6) = 91$; $f(0, 4) = 99$; $f(0, 2) = 99 = f(0, 4)$. Теперь берем меньший шаг и повторяем те же действия, приняв за исходную точку (0, 4), и т. д.

Иногда вместо метода покоординатного спуска используют его модификацию − более быстрый *метод Хука – Дживса*. Здесь выбирается начальная базисная точка $X^{(1)} = (x_1^{(1)}, x_2^{(1)}, ..., x_n^{(1)})$ и начальный шаг $H = (h_1, h_2, \ldots, h_n)$. На очередном этапе вычисляется значение функции в базисной точке и выясняется направление возрастания функции в ее окрестности. Для этого поочередно изменяем значения каждой переменной. Так из выбранной точки смещаемся «вправо» по x_1 на шаг h_1; если увеличение значений функции не наблюдается, возвращаемся в исходное положение и смещаемся «влево»; если и здесь нет увеличения функции, отступаем назад. Проводим аналогичные смещения по остальным координатам, получая новую базисную точку $X^{(2)}$.

Если $X^{(1)} = X^{(2)}$, уменьшаем величину шагов (компоненты вектора H) и повторяем вышеописанный поиск направления. Если шаг достаточно мал, принимаем базисную точку за оптимальное решение.

Если $X^{(1)} \neq X^{(2)}$, производим поиск максимума в выбранном направлении (так называемый *поиск по образцу*). Берем точку $Z = X^{(1)} + 2\,(X^{(2)} - X^{(1)})$ и производим аналогичный анализ направления локального (в ее окрестности) увеличения значений функции. Если значение функции в полученной точке больше значения в базисной точке $X^{(2)}$, то принимаем эту точку за очередную базисную и продолжаем поиск по образцу, в противном случае ведем анализ направления прироста в окрестности $X^{(2)}$.

Заметим, что оба эти метода хорошо работают для унимодальных функций и терпят крах для функций, имеющих локальные экстремумы (взобравшись на горку, мы не можем перебраться на другую, более высокую).

8.2.2. Градиентные методы

Выбираем начальную базисную точку $X^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})$ и начальный шаг h. Вычисляем значение функции в базисной точке и отыскиваем *градиент* функции

$$\text{grad } F(X^{(0)}) = \left\{ \frac{\partial F}{\partial x_1}, \quad \frac{\partial F}{\partial x_2}, \quad ..., \quad \frac{\partial F}{\partial x_n} \right\},$$

который определяет *направление наибольшего роста* функции в окрестности этой точки, нормируем его

$$\text{grad}_\text{н}\, F(X^{(0)}) = \text{grad } F(X^{(0)}) / |\text{grad } F(X^{(0)})|$$

и переходим в точку по этому направлению

$$X = X^{(0)} + h\, \text{grad}_\text{н}\, F(X^{(0)}).$$

Если $F(X) < F(X^{(0)})$, возвращаемся в базисную точку, уменьшаем шаг вдвое и повторяем переход по градиенту.

Если $F(X) > F(X^{(0)})$, принимаем эту точку за базисную и продолжаем переходы до тех пор, пока величина шага не станет меньше заданной погрешности.

Можно ускорить процесс «прогулок» по градиенту, если не перебирать шаг, а смещаться до точки максимума в направлении градиента, то есть отыскивать максимум функции $F(X^{(0)} + \lambda\, \text{grad}_\text{н}\, F(X^{(0)}))$ в диапазоне значений λ от 0 до h (поиск максимума функции одной переменной). Если найденный максимум достигается во внутренней точке диапазона, соответствующая точка принимается за базисную и процесс переходов продолжается. Если же он достигается при $\lambda = h$, можно удвоить шаг. Процесс переходов продолжают либо до «близости» смежных базисных точек, либо до получения слишком большого шага (признак неограниченности $F(X)$ по максимуму).

Поиск минимума функции, по существу, отличается лишь движением в направлении, обратном градиенту $X = X^{(0)} - h\, \text{grad}_\text{н}\, F(X^{(0)})$.

Естественно, и этот метод, называемый *методом наискорейшего спуска*, не дает гарантии глобальности найденного экстремума (все зависит от удачного выбора начальных точки и величины шага).

Возьмем для примера функцию $f(x, y, z) = -x^2 + 2\,x\,y - y^3 + y - (z-3)^2$, начальную точку $(0, 0, 0)$ и начальный шаг $h = 1$. Значение функции в базисной точке равно -9 и градиент определяется вектором

$$\text{grad}\, F(X) = (-2\,x + 2\,y,\ 2\,x - 3\,y^2 + 1,\ -2\,(z-3)) = (0, 1, 6).$$

Нормированный градиент здесь определяется вектором

$$\left(0, \frac{1}{\sqrt{37}}, \frac{6}{\sqrt{37}}\right) \approx (0,\ 0.1644,\ 0.9864),$$

и мы отыскиваем при λ из диапазона от 0 до $h = 1$ максимум функции $f(0 + 0\,\lambda,\ 0 + 0.1644\,\lambda,\ 0 + 0.9864\,\lambda) = -(0.1644\,\lambda)^3 + 0.1644\,\lambda - (0.9864\,\lambda - 3)^2$. Поскольку здесь максимум достигается при $\lambda = h = 1$, удваиваем h и отыскиваем максимум функции при λ от 0 до $h = 2$. Так как и здесь максимум достигается при $\lambda = h = 2$, удваиваем h и отыскиваем максимум функции при λ от 0 до $h = 4$. Здесь мы обнаруживаем факт, что максимум функции достигнут при $\lambda \approx 3.0616$ и равен 0.3754. Принимаем точку (0, 0.5033, 3.0200) за базовую и т. д. (см. таблицу).

N	Базовая точка			$f(x, y, z)$	h	λ	$\text{grad}_н\, F(X)$		
	x	y	z						
1	0.0000	0.0000	0.0000	-9.0000	1	1.0000	0.0000	0.1644	0.9864
2					2	2.0000			
3					4	3.0616			
4	0.0000	0.5033	3.0200	0.3754	4	0.8729	0.9720	0.2317	–0.0386
5	0.8485	0.7056	2.9863	0.8316	4	0.2201	–0.2310	0.9727	0.0221
6	0.7976	0.9197	2.9912	0.9726	4	0.1932	0.9700	0.2293	0.0702
7	0.9852	0.9640	3.0047	0.9970	4	0.0287	–0.2262	0.9728	–0.0505
8	0.9788	0.9919	3.0033	0.9997	4	0.0200	0.9495	0.2055	–0.2370
9	0.9977	0.9960	2.9985	1.0000	4	0.0030	–0.1728	0.9739	0.1473

8.3. Оптимизация при ограничениях. Множители Лагранжа

В случае нескольких переменных необходимым условием для точки экстремума функции $F(X)$ является обращение в нуль ее частных производных. Что касается типа экстремума, то здесь на помощь приходит, в частности, *правило Сильвестра*, согласно которому, для того чтобы в точке достигался минимум, необхо-

дима и достаточна положительность главных миноров матрицы вторых частных производных (функция выпукла в точке). Для достижения максимума необходимо и достаточно чередование знаков главных миноров этой матрицы, начиная со знака минус (признак вогнутости функции).

Например, для функции $F(x, y, z) = -x^2 + 2xy - y^3 + y - (z-3)^2$ ищем точки экстремума, решая систему

$$\frac{\partial F}{\partial x} = -2x + 2y = 0,$$
$$\frac{\partial F}{\partial y} = 2x - 3y^2 + 1 = 0,$$
$$\frac{\partial F}{\partial z} = -2(z-3) = 0,$$

откуда обнаруживаем две точки экстремума $\left(-\frac{1}{3}, -\frac{1}{3}, 3\right)$ и (1, 1, 3). Для определения типа экстремума по Сильвестру строим матрицу значений вторых производных в этих точках

$$\begin{bmatrix} \frac{\partial^2 F}{\partial x^2} & \frac{\partial^2 F}{\partial x \partial y} & \frac{\partial^2 F}{\partial x \partial z} \\ \frac{\partial^2 F}{\partial y \partial x} & \frac{\partial^2 F}{\partial y^2} & \frac{\partial^2 F}{\partial y \partial z} \\ \frac{\partial^2 F}{\partial z \partial x} & \frac{\partial^2 F}{\partial z \partial y} & \frac{\partial^2 F}{\partial z^2} \end{bmatrix} = \begin{bmatrix} -2 & 2 & 0 \\ 2 & -6y & 0 \\ 0 & 0 & -2 \end{bmatrix} = \begin{cases} \begin{bmatrix} -2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & -2 \end{bmatrix} \\ \begin{bmatrix} -2 & 2 & 0 \\ 2 & -6 & 0 \\ 0 & 0 & -2 \end{bmatrix} \end{cases}.$$

Легко увидеть, что в точке (1, 1, 3) главные миноры равны соответственно −2, 8, −16, т. е. в этой точке достигается максимум. Что касается второй точки, то здесь главные миноры равны −2, −8, 16 и говорить приходится лишь об экстремуме по той или иной переменной (максимум по z и своеобразное «седло» по x и y).

Рассмотрим поиск экстремума функции $F(X) = F(x_1, x_2, \ldots, x_n)$ при условиях в форме $G_k(X) = 0$, $k = 1, 2, \ldots, m$. Составим так называемую функцию Лагранжа

$$L(X, \Lambda) = F(X) + \sum_{k=1}^{m} \lambda_k G_k(X).$$

Можно доказать, что *необходимым условием наличия экстремума для поставленной задачи является обращение в нуль частных производных функции Лагранжа*, то есть выполнение условий

$$\frac{\partial L}{\partial X} = \frac{\partial F}{\partial X} + \sum_{k=1}^{m} \lambda_k \frac{\partial G_k(X)}{\partial X} = 0,\ \frac{\partial L}{\partial \lambda_k} = G_k(X) = 0,\ k = \overline{1,m}$$

(заметьте, что вопрос о типе экстремума здесь остается открытым).

Так при поиске экстремумов функции $f(x, y) = (x^3 + y^3) / 3$ при условии $x + y = 1$ строим функцию Лагранжа

$$L(x, y, \lambda) = (x^3 + y^3) / 3 + \lambda\,(x + y - 1)$$

и решаем систему уравнений для частных производных

$$x^2 + \lambda = 0;\ y^2 + \lambda = 0;\ x + y - 1 = 0.$$

Откуда получаем $x = y = 0.5,\ \lambda = -0.25$. Использовав правило Сильвестра для $f(x, y)$, обнаруживаем, что найдена точка минимума.

В случае, когда ограничения задачи допускают неравенства, например $G_k(X) \le 0$, $k = 1, 2, \ldots, m$, вводим так называемые *ослабляющие переменные* u_k^2 так, что $G_k(X) + u_k^2 = 0$, $k = 1, 2, \ldots, m$, строим функцию Лагранжа

$$L(X, \Lambda, U) = F(X) + \sum_{k=1}^{m} \lambda_k [G_k(X) + u_k^2]$$

и записываем необходимые условия экстремума задачи в виде

$$\frac{\partial L}{\partial X} = \frac{\partial F}{\partial X} + \sum_{k=1}^{m} \lambda_k \frac{\partial G_k(X)}{\partial X} = 0,$$

$$\frac{\partial L}{\partial \lambda_k} = G_k(X) + u_k^2 = 0,\ k = \overline{1,m},$$

$$\frac{\partial L}{\partial u_k} = 2\lambda_k u_k = 0, k = \overline{1,m}.$$

Умножив третью группу условий на u_k, получаем

$$u_k \frac{\partial L}{\partial u_k} = 2\lambda_k u_k^2 = -2\lambda_k G_k(X) = 0, k = \overline{1,m}.$$

Откуда видно, что выполнение ограничения в форме неравенства (оптимум не на границе) требует обращения соответствующего λ в нуль.

8.4. Условия Куна – Таккера

Пусть стоит задача минимизации функции $F(X) = F(x_1, x_2, \ldots, x_n)$ при условиях $G_k(X) \le 0$, $k = 1, 2, \ldots, m$, $X \ge 0$. Возьмем функцию Лагранжа в виде

$$L(X, \Lambda) = F(X) + \sum_{k=1}^{m} \lambda_k G_k(X).$$

В предположении регулярности, то есть существования хотя бы одной точки, в которой $G_k(X) < 0$, $k = 1, 2, \ldots, m$, можно доказать, что существует точка (X^*, Λ^*) такая, что $X^* \ge 0$, $\Lambda^* \ge 0$ и $L(X^*, \Lambda) \le L(X^*, \Lambda^*) \le L(X, \Lambda^*)$ при всех $X \ge 0$, $\Lambda \ge 0$. Это утверждение называют *теоремой Куна – Таккера*[*] или *теоремой о седловой точке* (в указанной точке функция Лагранжа достигает минимума по X и максимума по Λ). Если присутствующие здесь функции дифференцируемы, условия Куна – Таккера можно записать и в дифференциальной форме в окрестности седловой точки:

$$\frac{\partial L}{\partial x_j} \ge 0,\ x_j \frac{\partial L}{\partial x_j} = 0,\ x_j \ge 0,\ j = \overline{1, n},$$
$$\frac{\partial L}{\partial \lambda_k} \le 0,\ \lambda_k \frac{\partial L}{\partial \lambda_k} = 0,\ \lambda_k \ge 0, k = \overline{1, m}.$$

Пример 1. Пусть требуется минимизировать функцию

$$2 x_1 - 2 x_2 - x_1^2 - 2 x_1 x_2 + 2 x_2^2$$

при условиях

$$4 x_1 + 3 x_2 \le 12,$$
$$x_2 \le 3,$$
$$x_1 \ge 0, x_2 \ge 0.$$

[*] Работа над теоремой о седловой точке была начата Алом Таккером и его студентами Х. Куном и Д. Джейлом в 1948 году, но, по утверждению Д. Данцига, создателем теоремы является Джон фон Нейман, а Таккера и его коллег ценят как авторов строгого доказательства (1951).

Берем функцию Лагранжа $L(X, \Lambda) = 2\,x_1 - 2\,x_2 - x_1^2 - 2\,x_1\,x_2 + 2\,x_2^2 + \lambda_1\,(4\,x_1 + 3\,x_2 - 12) + \lambda_2\,(x_2 - 3)$ и записываем условия Куна – Таккера

$$\frac{\partial L}{\partial x_1} = 2 - 2x_1 - 2x_2 + 4\lambda_1 \geq 0,\ x_1 \frac{\partial L}{\partial x_1} = 0,\ x_1 \geq 0,$$

$$\frac{\partial \mathrm{L}}{\partial x_2} = -2 - 2x_1 + 4x_2 + 3\lambda_1 + \lambda_2 \geq 0,\ x_2 \frac{\partial L}{\partial x_2} = 0,\ x_2 \geq 0,$$

$$\frac{\partial L}{\partial \lambda_1} = 4x_1 + 3x_2 - 12 \leq 0,\ \lambda_1 \frac{\partial L}{\partial \lambda_1} = 0,\ \lambda_1 \geq 0,$$

$$\frac{\partial \mathrm{L}}{\partial \lambda_2} = x_2 - 3 \leq 0,\ \lambda_2 \frac{\partial L}{\partial x_2} = 0,\ \lambda_2 \geq 0.$$

Последующая работа по существу сводится к перебору допустимых решений для этой системы условий. Допустим, что λ_1 и λ_2 равны нулю, а x_1 и x_2 отличны от нуля. Тогда производные по x_1 и x_2 должны обратиться в нуль:

$$2 - 2\,x_1 - 2\,x_2 = 0,$$
$$-2 - 2\,x_1 + 4\,x_2 = 0.$$

Откуда получаем $x_1 = 1/3$ и $x_2 = 2/3$. Так как это решение отвечает остальным условиям, то мы получили точку минимума для исходной задачи (внутренняя точка множества допустимых решений).

Пример 2. Пусть необходимо минимизировать функцию

$$-2\,x_1 - 2\,x_2 - x_1^2 - 2\,x_1\,x_2 + 2\,x_2^2$$

при условиях примера 1. Берем функцию Лагранжа $L(X, \Lambda) = -2\,x_1 - 2\,x_2 - x_1^2 - 2\,x_1\,x_2 + 2\,x_2^2 + \lambda_1\,(4\,x_1 + 3\,x_2 - 12) + \lambda_2\,(x_2 - 3)$ и записываем условия Куна – Таккера

$$\frac{\partial L}{\partial x_1} = -2 - 2x_1 - 2x_2 + 4\lambda_1 \geq 0,\ x_1 \frac{\partial L}{\partial x_1} = 0,\ x_1 \geq 0,$$

$$\frac{\partial L}{\partial x_2} = -2 - 2x_1 + 4x_2 + 3\lambda_1 + \lambda_2 \geq 0,\ x_2 \frac{\partial L}{\partial x_2} = 0,\ x_2 \geq 0,$$

$$\frac{\partial L}{\partial \lambda_1} = 4x_1 + 3x_2 - 12 \leq 0,\ \lambda_1 \frac{\partial L}{\partial \lambda_1} = 0,\ \lambda_1 \geq 0,$$

$$\frac{\partial \mathrm{L}}{\partial \lambda_2} = x_2 - 3 \leq 0,\ \lambda_2 \frac{\partial L}{\partial x_2} = 0,\ \lambda_2 \geq 0.$$

Аналогичное допущение о равенстве нулю λ_1 и λ_2 и отличных от нуля x_1 и x_2 приводит к системе

$$-2 - 2\,x_1 - 2\,x_2 = 0,$$
$$-2 - 2\,x_1 + 4\,x_2 = 0,$$

решение которой $x_1 = -1$ и $x_2 = 0$ не удовлетворяет ограничениям, и выдвинутая гипотеза отвергается.

Допустим, что λ_1 и λ_2 отличны от нуля. Тогда производные по λ_1 и λ_2 должны обратиться в нуль:

$$4x_1 + 3x_2 - 12 = 0,$$
$$x_2 - 3 = 0.$$

Откуда находим $x_1 = 0.75$ и $x_2 = 3$, что ведет к равенству нулю частных производных по x

$$\frac{\partial L}{\partial x_1} = -2 - 2x_1 - 2x_2 + 4\lambda_1 = 0,$$

$$\frac{\partial L}{\partial x_2} = -2 - 2x_1 + 4x_2 + 3\lambda_1 + \lambda_2 = 0.$$

Откуда получаем значение $\lambda_2 < 0$, что противоречит условиям.

Допустим, что x_1 и λ_1 отличны от нуля. Тогда

$$\frac{\partial L}{\partial x_1} = -2 - 2x_1 - 2x_2 + 4\lambda_1 = 0,$$

$$\frac{\partial L}{\partial \lambda_1} = 4x_1 + 3x_2 - 12 = 0.$$

Откуда находим $x_1 = 3$ и $\lambda_1 = 2$, что согласуется с остальными условиями. Таким образом, получено оптимальное решение $x_1 = 3$ и $x_2 = 0$.

Приведенные примеры показывают, что использование условий Куна – Таккера для решения задач оптимизации − тяжкий труд, но пусть критикует тот, кто предложит что-то менее трудоемкое.

Заметим, что на основе этих условий строится достаточно простой и быстрый метод поиска экстремума линейной функции при линейных ограничениях − *симплексный метод* решения задач *линейного программирования*. Простота решения здесь объясняется тем, что в геометрической интерпретации множество решений представляет собой выпуклый многогранник и экстремумы целевой функции могут достигаться лишь в вершинах этого многогранника, но не внутри его.

На этих же условиях строятся и *методы выпуклого программирования* (поиска максимума вогнутой или минимума выпуклой функции на выпуклом множестве допустимых решений). Рассмотрению этих методов посвящена обширная литература по *математическому программированию* (поиску экстремумов функций при наличии ограничений).

8.5. Оптимизация с ограничениями. Методы штрафных функций

Пусть стоит задача минимизации функции $F(X)$ при условиях $G_k(X) \geq 0$, $k = 1, 2, \ldots, m$. Возьмем функцию $Z(X, r) = F(X) + R(X)$, где $R(X) \to \infty$ (принимает достаточно большие значения) при подходе к границе области и называется *штрафной функцией*.

Например, если взять

$$R(X) = r\sum_{k=1}^{m}\frac{1}{G_k(X)} \text{ или } R(X) = -r\sum_{k=1}^{m}\ln G_k(X),$$

где $r > 0$ выбирается так, чтобы точка минимума $Z = Z(X, r)$ незначительно отличалась от точки минимума исходной функции $F(X)$. Априорной оценки для r в общем случае получить не удается, и потому сначала ищут точку минимума $Z = Z(X, r)$ при некотором заданном r, затем при меньшем (вдвое или вдесятеро) значении и т. д. до тех пор, пока очередные оценки координат точки минимума не станут достаточно близкими. Поиск минимума здесь можно проводить методом наискорейшего спуска, приняв за начальную любую точку, удовлетворяющую ограничениям. Очевидно, что при разумном подходе к выбору шага «прыжок через барьер» невозможен. Таким образом, метод штрафных функций сводит решение задачи оптимизации с ограничениями к задаче оптимизации без ограничений. Рассмотренный подход относят к *методам внутренних штрафных*, или *барьерных функций*.

Идея *метода внешних штрафных функций* связана с выбором

$$R(X) = r \max^2(0, -G_k(x)), \text{ где } r \to \infty,$$

и возможностью принять на начальную любую точку, не удовлетворяющую ограничениям.

Рассмотрим для иллюстрации метода задачу минимизации $F(x) = x^2 - 4x$ при условии $G(x) = 1 - x \geq 0$. Берем $R(x) = r / (1 - x)^2$. На графике $Z(X, r)$ легко увидеть, что точки ее минимумов при $r \to 0$ сходятся к решению поставленной задачи (рис. 8.2).

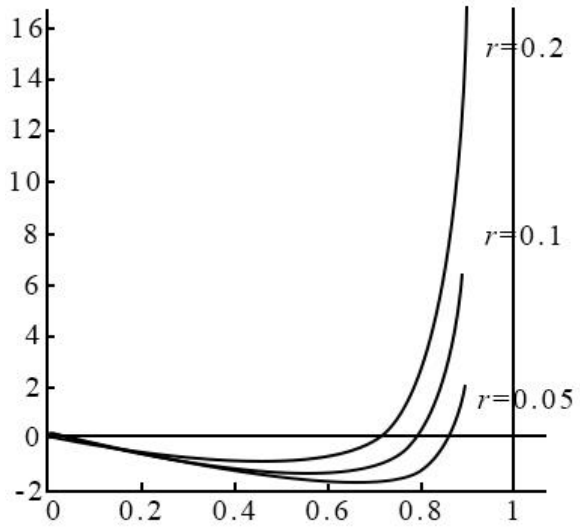


Рис. 8.2

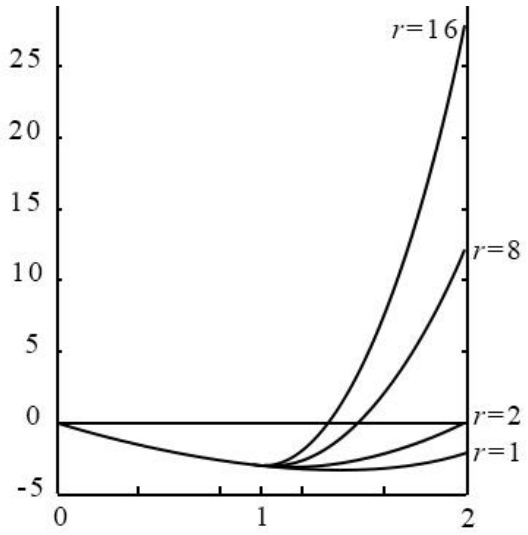


Рис. 8.3

Возьмем $R(x) = r \max^2(0, -(1 - x))$. Здесь $Z(X, r) = F(X) + r \max^2(0, -(1 - x))$ совпадает с $F(x)$ на множестве допустимых точек. Отыскав из

$$\frac{\partial Z}{\partial x} = 2x - 4 + 2r \max(0, x - 1) = 0$$

точку минимума $x(r) = \dfrac{2 + r}{1 + r} \to 1$, $r \to \infty$, на рис. 8.3 видим процесс поиска методом внешней штрафной функции (начинается от точки, не удовлетворяющей ограничениям).

8.6. Оптимизация с ограничениями. Градиентные методы

Существует исключительное многообразие модификаций градиентных методов. Остановимся только на двух подходах к реализации идеи «градиентного спуска», которые часто используются при разработке программных модулей в системах программирования и имеют многочисленные ссылки в литературе.

8.6.1. Метод проектируемого градиента Д. Розена

Пусть стоит задача минимизации $F(X)$ при условиях

$$G_k(X) \le 0,\ k = 1, 2, \ldots, m,$$
$$G_k(X) = 0,\ k = m + 1, m + 2, \ldots, K.$$

Метод проектируемого градиента является обобщением метода наискорейшего спуска, т е. использует идею смещения от выбранной допустимой точки в направлении (рис. 8.4), обратном градиенту (по антиградиенту). Если же точка лежит на границе (линии или поверхности) множества допустимых решений, то такое смещение невозможно из-за выхода за пределы множества. Поэтому антиградиент проецируют на касательную плоскость к множеству в выбранной точке, смещаются в этой плоскости на выбранный шаг и затем полученную новую точку, не являющуюся допустимой, проецируют на границу множества. Эти действия осуществимы с относительной простотой лишь в случае линейных ограничений (граничные поверхности плоскости или прямые линии).

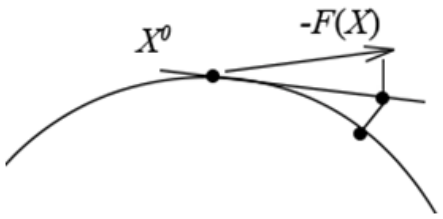


Рис. 8.4

В общем же случае берут приемлемое малое $\varepsilon > 0$ и исходные условия заменяют условиями, несколько расширяя исходное множество:

$$G_k(X) \le \varepsilon,\ k = 1, 2, \ldots, m,$$
$$|G_k(X)| \le \varepsilon,\ k = m + 1, m + 2, \ldots, K.$$

Затем выбирается некоторая точка X^0, удовлетворяющая приведенным условиям, и выясняется множество индексов граничных поверхностей, содержащих указанную точку, $C = \left\{ k / \left| G_k(X^0) \right| \le \varepsilon \right\}$. Находим многогранник, образуемый касательными плоскостями:

$$G_k(X) \cong G_k(X^0) + \sum_{j=1}^{n} \frac{\partial G_k(X^0)}{\partial x_j}(x_j - x_j^0) \le 0,\ k \le m,\ k \in C,$$

$$G_k(X) \cong G_k(X^0) + \sum_{j=1}^{n} \frac{\partial G_k(X^0)}{\partial x_j}(x_j - x_j^0) = 0,\ k > m.$$

Если обозначить через A_k вектор частных производных функции $G_k(X^0)$ и учесть $\left| G_k(X) \right| \le \varepsilon$ для $k \in C$, то при проецировании на многогранник вектора смещений $p_j = x_j - x_j^0$, $j = 1, 2, \ldots, n$ должны соблюдаться условия (в векторной записи)

$$(A_k, p) \le 0,\ k \le m,\ k \in C;\ (A_k, p) = 0,\ k > m.$$

Чтобы выбрать из всех таких векторов тот, который ближе к антиградиенту целевой функции, потребуем максимума скалярного произведения $(-F(X^0), P)$ или минимума $(F(X^0), P)$. Решение этой частной оптимизационной задачи дает нам оптимальное направление для перехода в точку минимума $F(X)$.

Если исходная точка не лежала точно на границе, то проецируем ее на многогранник, для чего решаем задачу минимизации $(X - X^0, X - X^0)$ при условии $G_k(X^0) + (A_k, p) = 0$, $k \in C$. Решение этой задачи с помощью множителей Лагранжа можно представить в виде

$$\overline{X}^0 = X^0 + A(A^T A)^{-1} G(X^0),$$

где A – матрица из векторов A_k, $k \in C$; G – вектор значений соответствующих $G_k(X^0)$. Теперь остается на выбранном направлении, не выходя за пределы установленной окрестности границы множества, найти точку минимума $F(X)$, для чего решаем задачу минимизации $F(\overline{X}^0 + \lambda P)$ при условиях

$$G_k(\overline{X}^0 + \lambda P) \le \varepsilon,\ k = 1, 2, \ldots, m,$$

$$\left| G_k(\overline{X}^0 + \lambda P) \right| \le \varepsilon,\ k = m + 1, m + 2, \ldots, K,\ \lambda \ge 0.$$

8.6.2. Метод возможных направлений Г. Зойтендейка

Метод возможных направлений Г. Зойтендейка, в отличие от метода проектируемых градиентов, предназначен для поиска экстремума при наличии ограничений только типа неравенств. Если в методе Розена допускались переходы к любым точкам в окрестности границы, то здесь допускается переход к точкам, строго удовлетворяющим ограничениям. Несколько отличается и подход в выборе направления перехода: если метод Розена ориентировался на градиент в окрестности точки, метод Зойтендейка пытается учесть величину смещения по ограничениям.

Пусть стоит задача минимизации $F(X)$ при условиях $G_k(X) \le 0,\ k = 1, 2, \ldots, m$ и выбраны некоторая точка X^0, удовле-

творяющая приведенным условиям, положительное значение ε и $C = \{\ k / -\varepsilon < G_k(X^0) \le 0\}$

То есть C определяет множество границ, в окрестности которых лежит выбранная точка и присутствует опасность выхода за пределы множества. *Допустимые* направления P (рис. 8.5), не выводящие при малых смещениях за пределы ограничений, должны составлять с нормалью к поверхности ограничения тупой угол, т. е. для них должны выполняться условия

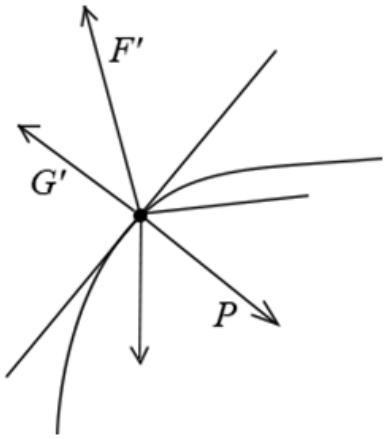


Рис. 8.5

$$(G_k(X^0), P) < 0,\ k \in C.$$

Если к тому же найдутся среди допустимых направления, составляющие с антиградиентом острый угол, т. е. $(F(X^0), P) < 0$, то эти направления являются *подходящими*.

Зойтендейк предлагает искать очередное приближение в виде $X^1 = X^0 + \lambda P$, где P – нормированный вектор, удовлетворяющий поставленным выше условиям и обеспечивающий минимум максимальной из величин $(F(X^0), P)$ и $(G_k(X^0), P)$, $k \in C$. Если этот минимум меньше $-\varepsilon$, то отыскивается λ, дающее минимум функции $F(X)$ на выбранном направлении, т. е. решается одномерная задача минимизации $F(X^0 + \lambda P)$ при условиях $G_k(X^0 + \lambda P) \le 0$, $k = 1, 2, \ldots, m$.

Если же этот минимум больше $-\varepsilon$, то можно утверждать, что найдена точка минимума с точностью ε (если необходимо, можно уменьшить ε и продолжить поиск).

Более подробное изложение специфики реализации градиентных методов можно найти в литературе. Здесь мы ограничились лишь описанием их идеологии, откуда читатель может сделать соответствующие выводы. Кстати, при решении задач большой размерности существенную помощь (по крайней мере, для выбора начальных оценок) можно получить, используя метод Монте-Карло (см. п. 6.6).

8.7. Оптимизация функций средствами MatLab

Если в случае решения задач линейной алгебры MatLab предоставляет исключительную библиотеку функций, то в области оптимизации такая библиотека выглядит несколько скромнее (особенно для задач с ограничениями).

Небесполезна при оптимизации функция `fzero('f(x)',x0)` – поиск нуля функции одной переменной (`x0` – начальное приближение). Так команда `fzero('x^2-5*x+4', 15)` оперативно дает ответ `4.0000`. Алгоритм работы функции сочетает методы дихотомии, хорд и обратной квадратической интерполяции (метод Форсайта).

Для систем двух уравнений определенную помощь в поиске начального приближения может оказать функция `contour(Z)`, где `Z` – двумерный массив значений, или `contour(x,y,Z)`, где `x`, `y` – векторы аргументов для `Z`. Например,

```
» [x,y]=meshgrid(-2:0.2:2,-2:0.2:2); % задание
                                     поля аргументов;
» Z=x.*exp(-x.^2-y.^2); % массив значений
                          функции;
» [C,h]=contour(x,y,Z); % вывод контура
                          (рис. 8.6);
» clabel(C,h) % вывод меток для линий уровня
```

Уже взглянув на картинку (рис. 8.6), видим местоположение экстремумов).

Например, при решении системы

$$x^2 + y^2 = 1,$$
$$\sin(x + y) = 0.1 + x$$

для поиска начального приближения возьмем функцию

$$F(x, y) = [x^2 + y^2 - 1]^2 + [\sin(x + y) - 0.1 - x]^2$$

и построим для нее линии уровня. Поскольку с очевидностью значения аргументов по абсолютной величине не превышают 1, строим аналогичную приведенной выше программу

```
» [x,y]=meshgrid(0:0.01:1,0:0.01:1);
» Z=(x.^2+y.^2-1).^2+(sin(x+y)-0.1-x).^2;
» [C,h]=contour(x,y,Z,'b-'); % (рис. 8.7);
» clabel(C,h)
```

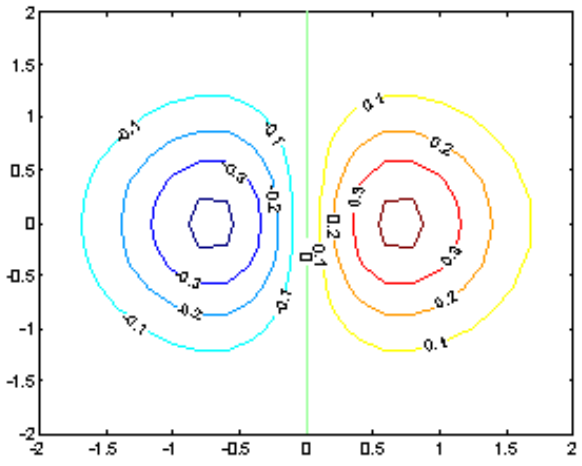


Рис. 8.6

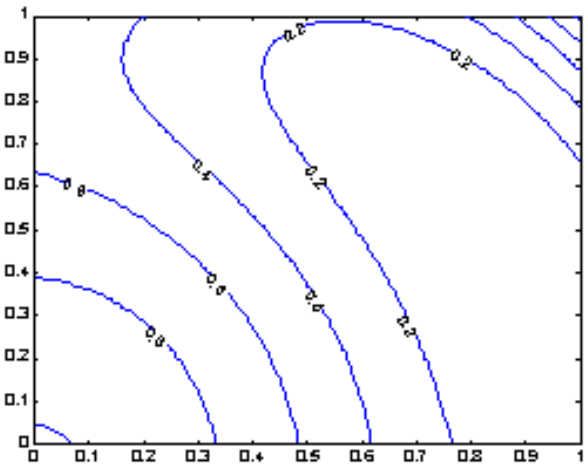


Рис. 8.7

Еще одно средство связано с функциями `fmin` и `fmins` (минимум функции одного или нескольких аргументов):

`fmin('имя функции',a,b)` – поиск на отрезке [a, b];

`fmin('имя функции',a,b,options)` – поиск с указанием режимов вывода промежуточных результатов (1 или 0), точности итераций по аргументу (по умолчанию 10^{-4});

`[xmin,options]=fmin('имя функции',a,b,options)` – в дополнение выводит текущие 18 режимов (точность для аргумента для функции, значение функции, число итераций, максимальное число итераций – по умолчанию 500).

Так задав

```
function f=fu1(x) / f=(sin(x+(1-y)^0.5)-0.1
                          -x)^2;/end
```

и выполнив `xmin=fmin('fu1',-1,1)`, получаем

`xmin=0.`**`8772`**`7194494488`.

По команде `xmin=fmin('fu1',-1,1,[1,1e-10])` выводится

```
1     –0.236068     0.651493        initial
2     0.236068      0.358526        golden
...
5     0.81966       0.00415883      golden
6     0.904252      0.00107623      parabolic
7     0.87853       2.14158e-006    parabolic
...
12    0.877286      1.70307e-021    parabolic
13    0.877286      2.3859e-016     parabolic
```

```
xmin = 0.87728629394929
```

`fmins('имя функции',x0)` – поиск локального минимума функции *n* переменных при начальном приближении `x0` (метод Нелдера – Мида).

`[xmin,options]=fmins('имя функции',x0, options)` – в дополнение выводит текущие режимы (максимальное число итераций $200 \cdot n$).

Например, для

```
function f=fu2(x) / f=(sin(x(1)+x(2))-0.1
                -x(1))^2+(x(1)^2+ x(2)^2-1)^2;
```

команда `fmins('fu2',[1,0])` дает

```
Maximum number of function evaluations (400)
has been exceeded (increase OPTIONS(14)).
ans = 0.96874999999999   0.02470312500000
```

(требуемая точность не достигнута).

ЦИТИРОВАННАЯ И РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Математический практикум / Г. Н. Положий, Н. А. Пахарева, И. З. Степаненко, П. С. Бондаренко, И. М. Великоиваненко ; под ред. Г. Н. Положего. − Москва : Физматгиз, 1960. – 512 с.

2. Березин, И. С. Методы вычислений (в 2-х томах). Т. 2 / И. С. Березин, Н. П. Жидков. − Москва : Физматгиз, 1962. – 620 с.

3. Крылов, А. Н. Лекции о приближенных вычислениях. − Москва : Гостехиздат, 1954. – 400 с.

4. Демидович, Б. П. Основы вычислительной математики / Б. П. Демидович, И. А. Марон. − Москва : Наука, 1966. – 664 с.

5. Хаусхолдер, А. С. Основы численного анализа. − Москва : Издательство иностранной литературы, 1956. – 321 с.

6. Фаддеев, Д. К. Вычислительные методы линейной алгебры / Д. К. Фаддеев, В. Н. Фаддеева. − Москва : Физматгиз, 1963. – 734 с.

7. Воеводин, В. В. Численные методы алгебры (теория и алгорифмы). − Москва : Наука, 1966. – 243 с.

8. Форсайт, Дж. Численное решение систем линейных алгебраических уравнений / Дж. Форсайт, К. Молер. − Москва : Мир, 1969. – 168 с.

9. Ланцош, К. Практические методы прикладного анализа. Справочное руководство. − Москва : Физматгиз, 1961. – 524 с.

10. Ланс, Дж. Н. Численные методы для быстродействующих вычислительных машин. − Москва : Издательство иностранной литературы, 1962. – 208 с.

11. Демидович, Б. П. Численные методы анализа: приближение функций, дифференциальные и интегральные уравнения / Б. П. Демидович, И. А. Марон, Э. З. Шувалова. − Москва : Наука, 1967. – 368 с.

12. Копченова, Н. В. Вычислительная математика в примерах и задачах / Н. В. Копченова, И. А. Марон. − Москва : Наука, 1972. – 368 с.

13. Агеев, М. И. Библиотека алгоритмов 151б-200б: Справочное пособие. Вып. 4 / М. И. Агеев, В. П. Алик, Ю. И. Марков ; под ред. М. И. Агеева. − Москва : Радио и связь, 1981. – 184 с.

14. Программное обеспечение ЭВМ МИР-1 и МИР-2. Численные методы. Т. 1. − Киев : Наукова думка, 1976. – 280 с.

15. Программное обеспечение ЭВМ МИР-1 и МИР-2. Программы. Т. 2. − Киев : Наукова думка, 1976. – 371 с.

16. Загускин, В. Л. Справочник по численным методам решения алгебраических и трансцендентных уравнений. − Москва : Физматгиз, 1960. – 216 с.

17. Воробьева, Г. Н. Практикум по вычислительной математике / Г. Н. Воробьева, А. Н. Данилова. − Москва : Высшая школа, 1990. – 208 с.

18. Бусленко, Н. П. Метод статистических испытаний (Монте-Карло) и его реализация на цифровых вычислительных машинах / Н. П. Бусленко, Ю. А. Шрейдер. − Москва : Физматгиз, 1961. – 228 с.

19. Годунов, С. К. Разностные схемы. Введение в теорию / С. К. Годунов, В. С. Рябенький. − Москва : Наука, 1977. – 440 с.

20. Яненко, Н. Н. Метод дробных шагов решения многомерных задач математической физики. − Новосибирск : Наука, 1967. – 197 с.

21. Хемминг, Р. В. Численные методы для научных работников и инженеров. − Москва : Наука, 1972. – 400 с.

22. Марчук, Г. И. Методы вычислительной математики. – Москва : Наука, 1977. – 456 с.

23. Тихонов, А. Н. Вводные лекции по прикладной математике / А. Н.Тихонов, Д. П. Костомаров. − Москва : Наука, 1984. – 192 с.

24. Мак-Кракен, Д. Численные методы и программирование на Фортране / Д. Мак-Кракен, У. Дорн. − Москва : Мир, 1977. – 583 с.

25. Самарский, А. А. Численные методы / А. А. Самарский, А. В. Гулин. − Москва : Наука, 1989. – 432 с.

26. Корн, Г. Справочник по математике для научных работников и инженеров / Г. Корн, Т. Корн. − Москва : Наука, 1984. – 832 с.

27. Беккенбах, Э. Ф. Современная математика для инженеров. − Москва : Издательство иностранной литературы, 1958. – 498 с.

28. Янке, Е. Специальные функции (формулы, графики, таблицы) / Е. Янке, Ф. Эмде, Ф. Леш. − Москва : Наука, 1964. – 344 с.

29. Банди, Б. Методы оптимизации (вводный курс). − Москва : Радио и связь, 1988. – 128 с.

Литература последних лет

30. Демидович, Б. П. Основы вычислительной математики / Б. П. Демидович, И. А. Марон. – Санкт-Петербург : Лань, 2011. – 672 с.

31. Демидович, Б. П. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения / Б. П. Демидович, И. А. Марон, Э. З. Шувалова. – Санкт-Петербург : Лань, 2010. – 400 с.

32. Копченова, Н. В. Вычислительная математика в примерах и задачах / Н. В. Копченова, И. А. Марон. – Санкт-Петербург : Лань, 2009. – 368 с.

33. Берцун, В. Н. Сплайны сеточных функций. – Томск : Изд-во Томск. гос. ун-та, 2002. – 124 с.

34. Меркулова, Н. Н. Методы приближенных вычислений : учеб. пособие / Н. Н. Меркулова, М. Д. Михайлов. – Томск : Изд-во Томск. гос. ун-та, 2011. – 184 с.

35. Самарский, А. А. Введение в численные методы : учеб. пособие для вузов. − Санкт-Петербург : Лань, 2005. – 288 с.

36. Бахвалов, Н. С. Численные методы в задачах и упражнениях : учеб. пособие / Н. С. Бахвалов, А. В. Лапин, Е. В. Чижонков. − Москва : Высшая школа, 2000. – 190 с.

37. Бахвалов, Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. − Москва : БИНОМ. Лаборатория знаний, 2009. – 632 с.

38. Чен, К. MATLAB в математических исследованиях / К. Чен, П. Джиблин, А. Ирвинг. – Москва : Мир, 2001. – 346 с.

39. Мэтьюз, Д. Г. Численные методы. Использование MATLAB / Д. Г. Мэтьюз, К. Д. Финк. − Москва : Вильямс, 2001. – 720 с.

41. Плис, А. И. MATHCAD 2000. Математический практикум / А. И. Плис, Н. А. Сливина. – Москва : Финансы и статистика, 2000. – 656 с.

42. Иглин, С. П. Математические расчеты на базе MATLAB. – Санкт-Петербург : БХВ – Петербург, 2005. – 640 с.

43. Потемкин, В. Г. Система инженерных и научных расчетов MATLAB 5.x. Т. 1. – Москва : Диалог-МИФИ, 1999. – 366 с.

44. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. − Санкт-Петербург : БХВ − Петербург, 2002. – 608 с.

45. Старченко, А. Б. Параллельные вычисления на многопроцессорных вычислительных системах / А. Б. Старченко, А. О. Есаулов. – Томск : Изд-во Томск. гос. ун-та, 2002. – 56 с.

46. Афанасьев, К. Е. Многопроцессорные вычислительные системы и параллельное программирование / К. Е. Афанасьев, С. В. Стуколов. – Кемерово : Кузбассвузиздат, 2003. – 233 с.

47. Недорезов, Л. В. Введение в экологическое моделирование : учеб. пособие. Т. 1. – Новосибирск : Изд-во Новосиб. гос. ун-та, 1998. – 142 с.

48. Недорезов, Л. В. Введение в экологическое моделирование : учеб. пособие. Т. 2. – Новосибирск : Изд-во Новосиб. гос. ун-та, 1999. – 110 с.

Тынкевич Моисей Аронович
Пимонов Александр Григорьевич

Введение в численный анализ

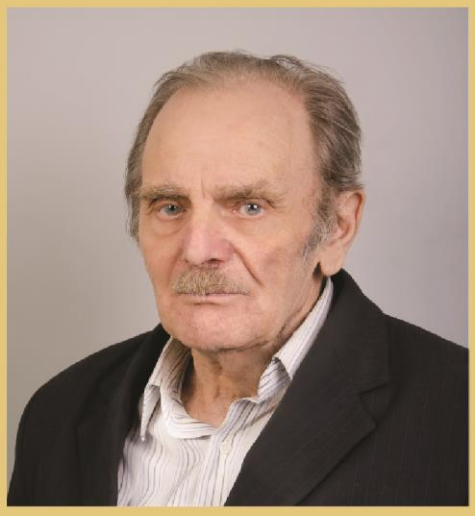
Учебное пособие

Редактор З. М. Савина

Подписано в печать 25.10.2017. Формат 60×84/16
Бумага офсетная. Гарнитура «Times New Roman». Уч.-изд. л. 11,5
Тираж 350 экз. Заказ № 109

КузГТУ, 650000, Кемерово, ул. Весенняя, 28

Издательский центр УИП КузГТУ, 650000, Кемерово, ул. Д. Бедного, 4А



Тынкевич Моисей Аронович – родился 28 февраля 1937 г. в г. Новосибирске, кандидат физико-математических наук, доцент, профессор кафедры прикладных информационных технологий. В 1959 г. окончил механико-математический факультет Томского государственного университета в группе двадцати четырех первых за Уралом выпускников по новой специальности «Вычислительная математика». В 1959 – 1966 гг. работал на кафедре вычислительной математики Томского государственного университета. С 1966 г. старший преподаватель кафедры экономики и организации производства Кузбасского политехнического института, с 1969 г. старший преподаватель новой в ВУЗе кафедры вычислительной техники и промэлектроники. Внес значительный вклад в становление и развитие кафедры прикладных информационных технологий. С 1998 г. является бессменным ответственным редактором научно-технического журнала «Вестник Кузбасского государственного технического университета». Подготовил более 80 научных работ и учебных пособий, несколько циклов методических разработок. Почетный работник высшего профессионального образования Российской Федерации (1997 г.). Награжден медалями «За особый вклад в развитие Кузбасса» (2001 г.), «За достойное воспитание детей» (2010 г.), «За веру и добро» (2015 г.). Ведет занятия по курсам «Численные методы», «Исследование операций и методы оптимизации», «Экономико-математические методы и модели», «Статистический анализ данных».

Пимонов Александр Григорьевич – родился 23 ноября 1959 г. в селе Чапаево Хакасской автономной области, доктор технических наук, профессор, заведующий кафедрой прикладных информационных технологий. В 1981 г. с отличием окончил факультет прикладной математики и кибернетики Томского государственного университета. В Кузбасском государственном техническом университете работает с 1985 г. (старший инженер, старший преподаватель, доцент, профессор, заместитель заведующего кафедрой, заведующий кафедрой, исполняющий обязанности декана факультета информационных технологий и менеджмента). Подготовил более 190 научных работ и учебно-методических разработок. Почетный работник высшего профессионального образования Российской Федерации (2010 г.), лучший профессор КузГТУ (2014 г.), лучший руководитель научно-исследовательской работы студентов КузГТУ (2014 г.), научный руководитель магистерской программы по направлению подготовки «Прикладная информатика» и программы подготовки кадров высшей квалификации по направлению «Информатика и вычислительная техника» (научная специальность «Математическое моделирование, численные методы и комплексы программ»). Ведет занятия по дисциплинам «Теория систем и системный анализ», «Математические и инструментальные методы поддержки принятия решений», «Математическое и имитационное моделирование», «Основы научных исследований», «Численные методы анализа».

ISBN 978-5-906969-35-4