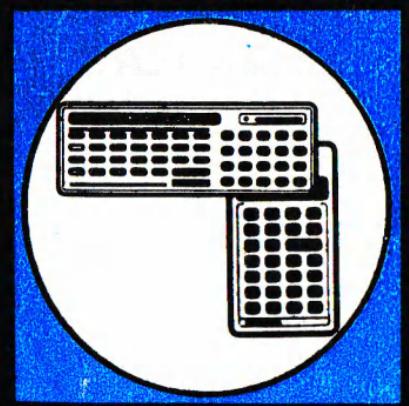


Г. Г. БАРАНОВСКАЯ
И. Н. ЛЮБЧЕНКО

**МИКРО-
КАЛЬКУЛЯТОРЫ
В КУРСЕ ВЫСШЕЙ
МАТЕМАТИКИ**



ПРАКТИКУМ



- Программируемые электронные микрокалькуляторы
- Элементы теории погрешностей
- Математические методы обработки данных



Г. Г. БАРАНОВСКАЯ
И. Н. ЛЮБЧЕНКО

МИКРО- КАЛЬКУЛЯТОРЫ В КУРСЕ ВЫСШЕЙ МАТЕМАТИКИ

ПРАКТИКУМ

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

Киев
Головное издательство издательского объединения
«Вища школа»
1987

22.19я73

Б24

УДК 621.38:51(076.5)

Микрокалькуляторы в курсе высшей математики: Практикум / Г. Г. Баарановская, И. Н. Любченко.— К.: Вища шк. Головное изд-во, 1987.— 288 с.

Описаны основные численные методы, предусмотренные программой курса: численные методы решения систем линейных и нелинейных уравнений, интерполяции и аппроксимации функций, статистической обработки экспериментальных данных, численного интегрирования, минимизации функций, решения дифференциальных уравнений.

Для каждого метода составлены программы в ко-дах микрокалькулятора «Электроника Б3-34», даны образцы решения контрольных примеров, задания для выполнения лабораторных работ и ответы к ним.

Для студентов высших технических учебных заведений.

Табл. 11. Ил. 32. Библиогр.: 37 назв.

Рецензенты: доктор физико-математических наук, профессор С. М. Белоносов (кафедра высшей математики Киевского инженерно-строительного института), кандидат физико-математических наук, доцент С. Д. Балашова (кафедра вычислительной математики Днепропетровского государственного университета).

Редакция литературы по информатике и автоматике
Зав. редакцией Г. Ф. Трофимчук

Б 2405000000—045
М211(04)—87 187—87

(C) Издательское объединение
«Вища школа», 1987

Меры по ускорению научно-технического прогресса предусматривают разработку новых поколений высокоеффективной техники, в частности электронно-вычислительных машин всех классов. Поэтому актуальным вопросом в процессе обучения стоит массовое овладение средствами вычислительной техники, программирования, широкая компьютеризация вузовского и школьного образования.

Математическая подготовка студентов предполагает развитие у них логического и алгоритмического мышления, в частности глубокое изучение основных численных методов и умение реализовать их на ЭВМ. Программа курса высшей математики предусматривает непрерывное использование вычислительной техники в учебном процессе: при решении задач по отдельным темам, при выполнении типовых расчетов, курсовых работ, а также при проведении математического лабораторного практикума. В процессе выполнения лабораторных работ у студентов вырабатывается умение выбирать оптимальный для данной задачи численный метод, получать решение с требуемой или возможной точностью, пользоваться вычислительными средствами, развиваются навыки правильной организации вычислений.

При этом широкое применение в качестве вычислительного средства индивидуального пользования получили малогабаритные микро-ЭВМ, называемые микрокалькуляторами. К основным достоинствам микро-ЭВМ следует отнести малую потребляемую мощность, портативность, простоту и безопасность в обращении, возможность активного диалога в процессе составления, редактирования, отладки программы и работы с ней.

Программируемые микрокалькуляторы, обладающие числовой и программной памятью, оказались достаточно эффективным вычислительным средством для проведения лабораторного практикума в курсе высшей математики. Выполнение сравнительно небольших расчетов, постоянно встречающихся в инженерной практике (решение уравнений и систем, интер-

поляция и аппроксимация функций, вычисление интегралов и численное дифференцирование, задачи оптимизации, решение дифференциальных уравнений, обработка экспериментальных данных и другие), можно успешно проводить на программируемых микрокалькуляторах.

Проведение лабораторного практикума по высшей математике на микро-ЭВМ способствует хорошему усвоению студентами алгоритмов численных методов, возможность активного диалога с машиной в процессе вычислений позволяет использовать ее в различных сферах деятельности. Квалифицированная работа с микрокалькулятором вырабатывает у пользователей определенную математическую культуру и алгоритмическую грамотность, что является предпосылкой для успешного овладения алгоритмическими языками, выполнения сложных расчетов на больших ЭВМ с использованием стандартных программ.

Настоящая книга является методическим пособием по лабораторному практикуму курса высшей математики на отечественных программируемых микрокалькуляторах. Пособие составлено в соответствии с программой курса высшей математики для вузов. При его написании использован опыт проведения лабораторных занятий кафедрами высшей математики Киевского политехнического института.

Создание данного пособия обусловлено тем, что многие вузы страны в настоящее время проводят математический практикум на программируемых микрокалькуляторах.

В предлагаемой книге по каждой лабораторной работе приводятся: краткие теоретические сведения по основным численным методам, оценка их погрешности, программы для микрокалькуляторов и инструкции по работе с ними, примеры решения контрольных примеров, варианты заданий для самостоятельных упражнений и ответы к ним.

Книга состоит из десяти глав. В гл. 1 приведена краткая история развития вычислительных средств, изложены основные сведения об отечественных программируемых МК «Электроника Б3-21», «Электроника МК-46», «Электроника Б3-34», «Электроника МК-54», «Электроника МК-56», «Электроника МК-61» и «Электроника МК-52», методика составления и записи программ для них, конкретные примеры составления программ.

В гл. 2 изложены элементы теории погрешностей. Остальные восемь касаются только лабораторных работ.

При подготовке к лабораторной работе студенты должны изучить соответствующий численный метод, ознакомиться с приведенной в пособии программой либо составить ее само-

стоятельно. На занятии каждый студент решает контрольный пример и выполняет вариант индивидуального задания, заполняет соответствующие таблицы результатов и проводит их анализ.

Методическое пособие написано для студентов и преподавателей высших технических учебных заведений. Оно может быть рекомендовано также учителям и учащимся средних специальных учебных заведений, инженерным и научным работникам, использующим программируемые микрокалькуляторы.

Глава 1. ПРОГРАММИРУЕМЫЕ ЭЛЕКТРОННЫЕ МИКРОКАЛЬКУЛЯТОРЫ

1.1. Краткая история развития вычислительных средств

Различные вспомогательные средства вычислений использовались с давних времен. К древнейшим счетным устройствам относятся китайские счеты суан-пан, счетная доска абак различных конструкций, применяемая древними греками, римлянами и в странах Древнего Востока, а также более поздняя разновидность абака — русские счеты, появившиеся около XVI века. Средства вычислений получили дальнейшее развитие в XVI веке, чему способствовали накопленные знания в области астрономии, открытие логарифмов. В 1614 г. шотландский математик Дж. Непер опубликовал таблицы логарифмов. Созданный им специальный прибор, известный под названием палочек Непера, предназначался для умножения многозначных чисел на однозначные. Этот прибор стал прообразом логарифмической линейки, изобретенной английским математиком У. Отредом в 20-е годы XVII века. Многократное усовершенствование логарифмической линейки позволило проводить на ней самые разнообразные вычисления: умножение, деление, возведение в степень, извлечение корня, вычисление логарифмов, значений тригонометрических функций. Она прослужила человечеству в качестве вычислительного средства индивидуального пользования более 350 лет.

Механические приборы для арифметических вычислений начали появляться в XVII веке. Первая модель вычислительной машины была разработана немецким ученым В. Шиккардом. Она предназначалась для автоматического выполнения четырех арифметических операций. В 1642 г. выдающийся французский ученый Б. Паскаль построил сущную машину для сложения и вычитания многозначных чисел. В 1694 г. известный немецкий математик и философ Г. В. Лейбниц сконструировал вычислительную машину для выполнения четырех арифметических действий — арифмометр. Из-за конструктивных недостатков и низкого уровня развития промышленности эти машины не получили широкого распространения. И только в 1874 г. петербургский инженер В. Т. Однер

разработал арифмометр, получивший всемирное признание. Арифмометры В. Т. Однера начали серийно выпускать в Петербурге в 1891 г. Они получили широкое распространение в разных странах, став основным средством механизации вычислительных работ в первой четверти XX века. Из других счетных устройств, предназначенных для выполнения основных арифметических действий, следует отметить арифмометр с непрерывной передачей десятков в единицу старшего разряда, изобретенный в 1878 г. знаменитым русским математиком П. Л. Чебышевым. Принцип, положенный в основу построения машины, был широко использован в разных странах при создании настольных механических клавишных вычислительных машин.

Значительный вклад в разработку теории математических машин внес выдающийся русский ученый А. Н. Крылов. В 1904 г. им создана первая в мире машина специального назначения для интегрирования обыкновенных дифференциальных уравнений.

Идея создания вычислительной машины с программным управлением принадлежит английскому ученому Ч. Бэббиджу. В 1833 г. он разработал проект машины, названной им аналитической, которая могла автоматически выполнять достаточно сложные вычисления по заранее составленной программе. Программа должна была задаваться с помощью отверстий, пробитых специальным образом в картонных картах (перфокартах). Первую в мире программу для вычисления чисел Бернулли на такой машине написала в 1843 г. леди Августа Ада Лавлейс, дочь известного английского поэта Дж. Г. Байрона. Проект Бэббиджа предусматривал наличие в аналитической машине внутренней механической памяти, устройств для выполнения арифметических действий, управления операциями, ручного набора исходных данных и выдачи окончательных результатов. Несмотря на то что этот проект, опередивший свое время, не был реализован, Ч. Бэббиджу принадлежит приоритет многих идей, положенных в основу первых электронных вычислительных машин (ЭВМ), созданных только в 40-х годах нашего века.

Первая действующая электромеханическая машина «Ц-3» с программным управлением была построена в 1941 г. немецким инженером К. Цузе. Более совершенная вычислительная машина на механических и релейно-контактных элементах «Марк-1» создана в 1944 г. под руководством американского физика Г. Эйкена.

Идея создания электронных (ламповых) вычислительных машин, приведшая к революционной перестройке всей вычис-

лительной техники, была высказана в 1937 г. американским физиком (болгарин по происхождению) Д. В. Атанасовым. Им в 1942 г. был практически построен электронный компьютер «ABC», однако ввести его в эксплуатацию помешала вторая мировая война. В 1946 г. американские физики Д. Мокли и Д. Эккерт создали первую действующую электронную машину «ЭНИАК».

В нашей стране первая ЭВМ МЭСМ была разработана в 1948–1951 гг. под руководством академика С. А. Лебедева в Институте электротехники АН УССР. Под его же руководством в 1952 г. была создана одна из первых быстродействующих вычислительных машин — БЭСМ.

За короткое время было создано несколько поколений ЭВМ. В машинах первого поколения (конец 40-х — середина 50-х годов) использовались электронно-вакуумные лампы, второго поколения (конец 50-х — начало 60-х годов) — полупроводниковые элементы, третьего поколения (середина 60-х годов) — интегральные схемы (ИС). На смену машинам третьего поколения приходят машины четвертого поколения, грань между которыми не очень четкая. В качестве элементной базы в них используются большие интегральные схемы (БИС). К ним относятся многомашинные, мультипроцессорные системы параллельной обработки, построенные на функционально модульной основе, а также современные персональные компьютеры.

Наряду с универсальными ЭВМ в начале 60-х годов получили широкое распространение «малые» вычислительные машины: микрокалькуляторы, микропроцессоры, микро- и мини-ЭВМ, персональные ЭВМ и ЭВМ специального назначения. Их появление неразрывно связано с новейшими достижениями микроэлектроники.

Первая в мире настольная электронная клавишная вычислительная машина (ЭКВМ) «Анита МК-8» была продемонстрирована на выставке в Лондоне в 1961 г. Серийный выпуск настольных ЭКВМ начался с 1964 г. («Вега» в нашей стране). К концу 60-х годов появились переносные ЭКВМ на интегральных схемах. В нашей стране с 1971 г. начат серийный выпуск «Искры».

Габаритные размеры и масса различных моделей ЭКВМ постепенно уменьшаются и в начале 70-х годов появились «карманные» электронные микрокалькуляторы (МК), а затем — инженерные МК с дополнительными регистрами памяти и подпрограммами для автоматического вычисления значений элементарных функций, а также сверхмалогабаритные программируемые МК. На них можно выполнять вычисления

по составленным пользователем программам в 60—100 шагов. Такие портативные цифровые устройства с автономным источником питания, предназначенные для выполнения инженерных и научных расчетов, пришли в наше время на смену логарифмической линейке.

В некоторых типах МК имеются устройства для хранения и ввода программы и данных (перфокарты, перфоленты, магнитные мини-карты, библиотечные сменные кассеты), миниатюрные встроенные знакопечатающие устройства. Однако все эти модификации ограничивают автономность, портативность и небольшую стоимость МК «карманного» типа. В настоящее время в нашей стране выпускаются разнообразные модели МК. Повышается качество их работы, надежность, уменьшаются габаритные размеры, масса, стоимость.

Микропроцессоры являются элементной базой для конструирования ЭВМ различного назначения. Они представляют собой стандартные логические блоки, выполняющие разнообразные функции под управлением хранимой в их памяти программы.

Микро-ЭВМ выполняют чаще всего функции программного управления автоматическими производственными линиями. Их встраивают в измерительную аппаратуру, станки, робототехнические устройства.

Мини-ЭВМ используются как вычислительные машины, на их основе создаются автоматизированные рабочие места для проведения инженерных и научных расчетов, автоматизации обработки экспериментальных данных, проектирования, конструирования и т. д.

1.2. Общие сведения о программируемых микрокалькуляторах

По функциональному назначению МК разделяются на простейшие, инженерные, программируемые и специализированные. Простейшие МК предназначаются для выполнения арифметических операций, более совершенные их модели имеют дополнительные регистры памяти для запоминания промежуточных результатов, а также позволяют, например, автоматически вычислять $1/x$, \sqrt{x} , %. На инженерных МК проводятся сложные расчеты инженерной и научной практики. Они имеют специальные встроенные микропрограммы для автоматического вычисления значений многих элементарных функций: прямых и обратных тригонометрических, десятичных и натуральных логарифмов, показательной и степенной. На

них можно осуществлять перевод градусов в радианы и наоборот, а также вычислять значения функций, микропрограммы которых отсутствуют.

На программируемых МК вычисления проводятся нажатием вручную на пульте соответствующих клавиш или по составленной пользователем и введенной в МК программе. Наличие в программируемых МК логических операторов условного и безусловного переходов, команд организации циклов и обращения к подпрограммам, косвенной адресации, программной памяти емкостью в несколько десятков шагов позволяет автоматизировать многие научные и инженерно-технические расчеты.

Специализированные МК предназначены для выполнения расчетов из некоторой узкой отрасли, например статистических, кассовых, и содержат соответствующие операторы вычисления специальных функций.

Более подробно рассмотрим основные типы отечественных программируемых МК, которые в последние годы находят широкое применение в учебном процессе, в частности, при проведении лабораторного практикума по численным методам курса высшей математики.

В нашей стране первая модель программируемого МК «Электроника Б3-21» была создана в 1977 году. МК имеет программную память емкостью 60 шагов и 14 регистров памяти. На его базе был разработан специализированный настольный программируемый МК «Электроника МК-46» (1980 г.), предназначенный для автоматического контроля различных производственных процессов. В нем предусмотрен ввод данных с внешних устройств.

Более совершенным по сравнению с ними является МК «Электроника Б3-34», выпускавшийся с 1980 года. Он имеет программную память в 98 шагов, более гибкую оперативную память, состоящую из четырех стековых регистров и дополненную регистром предыдущего результата, 14 адресуемых регистров памяти с произвольным доступом, переключатель для представления углов в градусах и радианах, большее число встроенных микропрограмм вычисления значений элементарных функций, а также, кроме прямой адресации, команды косвенных переходов и обращений, автоматической организации циклов.

На базе этой модели созданы аналогичные по функциональным возможностям и программно совместимые с ней настольный МК «Электроника МК-56», переносные «Электроника МК-54» (1982 г.), «Электроника МК-61» (1985 г.) и «Электроника МК-52» (1985 г.).

Для работы с МК необходимо предварительно ознакомиться с инструкцией по его эксплуатации. В настоящее время пока не созданы библиотеки типовых программ (прилагаемые к инструкции), что вынуждает пользователя программировать свою задачу самостоятельно.

Несмотря на различие в размерах и внешнем виде, все ЭВМ имеют общую структуру и принципы работы. Программируемые МК содержат: устройство ввода информации в память машины — клавиатуру; устройство вывода информации — индикатор; запоминающее устройство (ЗУ), состоящее из оперативной памяти для хранения исходных данных, промежуточных и окончательных результатов вычислений, запоминающего устройства для хранения программы вычислителя и постоянного запоминающего устройства для хранения встроенных (резидентных) программ автоматического вычисления значений элементарных функций; устройство управления переработкой информации совместно с арифметическим устройством (микропроцессор).

1.3. Ввод чисел

Микрокалькуляторы оперируют восьмиразрядными десятичными числами. Форма представления чисел может быть естественной с фиксированной запятой и экспоненциальной с плавающей десятичной запятой. Ввод чисел осуществляется с помощью цифровых клавиш: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; клавиши десятичного разделительного знака (запятой «,» или точки «.»), клавиши ввода порядка ВП и клавиши смены знака $/—/$. Числа с клавиатуры вводятся в операционный регистр X. Визуальный контроль вводимых с клавиатуры чисел, а также выводимых результатов вычислений осуществляется с помощью связанного с регистром X 12-разрядного индикатора: восемь разрядов мантиссы, два разряда порядка, два разряда знаков мантиссы и порядка. Диапазон представления чисел с фиксированной запятой от $\pm 10^{-7}$ до ± 99999999 ; с плавающей запятой от $\pm 10^{-7} \cdot 10^{99}$ до $\pm 99999999 \cdot 10^{99}$. Индикатор высвечивает содержимое регистра X.

Числа с фиксированной запятой вводятся, как при чтении, слева направо нажатием соответствующих цифровых клавиш, включая 0 для целой части дробного числа, запятой, если нужно; знак отрицательного числа $/—/$ набирается в конце. В МК «Электроника Б3-21» запятая занимает один разряд, поэтому дробные числа могут иметь семь значащих

цифр. В МК «Электроника Б3-34» запятая мантиссы отдельного разряда не занимает. Для чисел с плавающей запятой сначала вводится мантисса, если число отрицательное, то после ввода мантиссы нажимается клавиша $/\text{--}/$, затем нажимается клавиша порядка ВП, вводятся цифры порядка и его знак $/\text{--}/$, если он отрицательный.

При стандартной форме записи чисел запятая в мантиссе расположена после первой значащей цифры. Если числа вида $\pm m \cdot 10^{\pm n}$ при $m > 0$ и числа a при $|a| < 1$ пересылать из регистра Х в регистр У (нажатием клавиши \uparrow или В \uparrow) или в адресуемый регистр, то они автоматически переводятся в стандартную форму с плавающей запятой. Результаты вычислений высвечиваются на индикаторе в стандартной форме.

Приведем примеры ввода чисел нажатием соответствующих клавиш:

Число	Клавиши
25867	2 5 8 6 7
-125,6489	1 2 5 , 6 4 8 9 /--/
-0,0036	0 , 0 0 3 6 /--/
$2,56 \cdot 10^{-74}$	2 , 5 6 ВП 7 4 /--/
$-5,432 \cdot 10^{14}$	5 , 4 3 2 /--/ ВП 1 4

Вводить в МК числа можно в промежуточной форме, например, для ввода числа $125,146 \cdot 10^{-3}$ нажимаем клавиши 1 2 5 , 1 4 6 ВП 3 /--/. Если при вводе числа допущена ошибка, то нажимаем клавишу С_х очистки регистра Х и набираем число сначала. Если допущена ошибка при наборе числа порядка, то повторяем ввод порядка и его знака.

В МК предусмотрена блокировка ввода: если уже введено восемь цифр мантиссы, то нажатие цифровых клавиш не вызывает изменений на индикаторе.

Если в результате вычислений происходит переполнение разрядной сетки регистра Х или выполняется некорректная операция, на индикаторе МК «Электроника Б3-21» высвечивается 0 в первом разряде (вместо знака числа), а в МК «Электроника Б3-34» и построенных на его базе моделях высвечивается сигнал ЕГГОГ (от англ. *error — ошибка*).

1.4. Микрокалькуляторы «Электроника Б3-21» и «Электроника МК-46»

Внешний вид МК «Электроника Б3-21» представлен на рис. 1. В верхней части передней панели находится световой индикатор. Слева под ним расположен выключатель. МК может работать либо от расположенных внутри него аккумуляторов, либо от блока питания, включаемого в сеть 220 В. В первом случае необходимо повернуть выключатель вправо, во втором включение производится в такой последовательности: подключить к МК блок питания, включить блок в сеть, а затем повернуть выключатель. Выключение осуществляется в обратной последовательности. Следует запомнить, что повторное включение МК допускается не раньше, чем через 10 с после его выключения.

Оперативная память МК «Электроника Б3-21» состоит из двух операционных регистров X и Y с адресами соответственно 0 и 1, семи адресуемых регистров с адресами 2, ..., 8 и шести стековых регистров. Обозначим их соответственно RX, PY, P2, ..., P8, C1, ..., C6.

Числа вводятся с пульта в регистр X. Клавиша \uparrow предназначена для передачи содержимого регистра X в регистр Y. При этом содержимое RX сохраняется. Если содержимое RX является результатом выполнения некоторой операции (не вводилось с пульта), то при вводе нового числа содержимое RX автоматически заносится в PY. При нажатии клавиши $X \rightleftharpoons{} Y$ (в дальнейшем будем писать XY) происходит обмен содержимым в регистрах X и Y.

Вызов числа из адресуемого регистра с адресом N ($N = 2, \dots, 8$) в RX осуществляется нажатием клавиш FN. Для записи содержимого RX в регистр с адресом N нужно нажать клавиши PN. Адрес N набирается нажатием соответствующей цифровой клавиши.

В микрокалькуляторе «Электроника Б3-21» шесть стековых регистров C1, ..., C6 и регистр X образуют замкнутое кольцо, называемое *кольцевым стеком*. Схема кольцевого стека представлена на рис. 2.

Запись чисел в стековые регистры и их вызов осуществляется только через регистр X поворотом стека в двух направлениях — по часовой стрелке (нажатием клавиш Р Ø) или против часовой стрелки (нажатием клавиш Р Ø). При нажатии клавиш Р Ø содержимое регистра X записывается в регистр C1, числа в остальных регистрах стека передвигаются по часовой стрелке в соседний регистр стековой памяти, а содержимое регистра C6 записывается в регистр X. При нажатии



Рис. 1

клавиш РО передвижение содержимого регистров стека осуществляется аналогично, но уже против часовой стрелки. Вызов чисел из стековой памяти можно проводить в том порядке, в котором они заносились в стек, или в противоположном порядке.

Пример. Вычислить значение многочлена:

$$P(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + f \text{ при } x = x_0.$$

Вычисление проведем по схеме Горнера:

$$P(x) = (((((ax + b)x + c)x + d)x + e)x + f).$$

Запишем коэффициенты многочлена в стековую память по часовой стрелке:

(a) РО(b) РО(c) РО(d) РО(e) РО(f) РО

Число x_0 занесем в Р2: (x_0) Р2.

Здесь и далее запись (a) означает набор на цифровой клавиатуре числа a .

Тогда вычисление значения многочлена по схеме Горнера реализуется программой:

РО ↑ F2 × ↑ РО + С/П БП Р2

Для пятикратного повторения группы команд следует нажать клавиши В/О С/П и после ОСТАНОВов четыре раза — клавишу С/П.

Значение $P(x_0)$ высвечивается на индикаторе.

Клавиатура МК «Электроника Б3-21» содержит 29 клавиш. Некоторые из них имеют двойную или даже тройную символику — на клавише, выше и ниже ее. Операторы, обозначаемые верхними или нижними символами, вводятся в память после нажатия одной из префиксных клавиш Р или F соответствующего цвета.

Кроме рассмотренных выше клавиш ввода чисел и клавиш перемещения информации в регистрах памяти, на пульте имеются операционные клавиши (+, -, ×, ÷); клавиши для вычисления значений элементарных функций (X^y , \ln , e^{ix} , π , \sin , \cos , $1/X$, X^2 , \sqrt{x}); клавиши, используемые при программировании (безусловного — БП и условных — $X > 0$, $X \neq 0$, $X = 0$, $X < 0$ переходов, ПП — перехода к подпрограмме, В/О — возврата из подпрограммы в основную про-

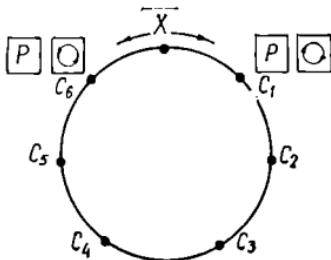


Рис. 2

грамму или перехода на нулевой адрес программы, ШГ, ШГ — сдвига счетчика адресов на шаг вперед или назад, НОП — стирание команды); клавиши С/П для пуска и ОСТАНОВа вычислений.

С помощью МК можно вычислять значения экспоненциальной функции мнимого аргумента $e^{ix} = \cos x + i \sin x$. При этом на индикаторе высвечивается значение действительной части $\cos x$, а в регистре У хранится мнимая часть $\sin x$. Поэтому, если после выполнения оператора e^{ix} нажать клавишу деления \div , то на индикаторе получим значение $\operatorname{tg} x$. Для вычисления значения $\operatorname{ctg} x$ можно нажать клавиши: $e^{ix} XU \div$.

Программная память состоит из счетчика адресов и массива из 60 регистров, образующих стек «карусельного» типа. Число регистров программной памяти определяет максимальное число команд (шагов) программы пользователя, одна команда занимает один регистр.

Программную память МК можно образно представить в виде книги [30], которая имеет десять страниц с номерами 0, ..., 9. На каждой странице размещается шесть строк с номерами 0, ..., 5. Каждая команда занимает одну строку (ячейку программной памяти). Ее положение определяется двузначным десятично-шестеричным числом — адресом команды. Первая цифра адреса указывает номер страницы, а вторая — номер строки. В программе адреса команд записываются подряд: 00, 01, 02, 03, 04, 05, 10, 11, ..., 95. Программа вводится в программную память МК с клавишного пульта управления. При вводе программы каждый оператор автоматически преобразуется в двузначный код. В режиме программирования при нажатии соответствующих клавиш код оператора заносится в очередной регистр памяти, а содержимое счетчика адресов после записи очередного шага программы автоматически увеличивается на единицу. При этом на индикаторе справа в 11-м и 12-м разрядах высвечивается адрес следующей команды, а слева — коды трех последних команд, что можно использовать при проверке и отладке программы.

Каждой команде программы, кроме ее адреса, отвечает адрес перехода. Адрес перехода представляет собой символ оператора, код которого на единицу больше адреса команды, на которую требуется перейти. В табл. 1.1 приведены коды команд и символов МК «Электроника Б3-21». Табл. 1.1 можно использовать для определения кодов команд при записи программ и для нахождения адреса перехода. Например, для перехода на команду с адресом 34 находим по таблице код

Таблица 1.1

Вторая цифра кода	Первая цифра кода									
	0	1	2	3	4	5	6	7	8	9
1	P 0	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9
2	F 0	F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8	F 9
3	P ↑	P XY	P X	P ÷	P ,	P /—/	P BΠ	P Cx	P —	P +
4	0	1	2	3	4	5	6	7	8	9
5	F ↑	F XY	F X	F ÷	F ,	F /—/	F BΠ	F Cx	F —	F +
6	↑	XY	X	÷	,	/—/	BΠ	Cx	—	+
7							F B/O	F BΠ	F ΠΠ	
8							XY	B/O	BΠ	C/Π
9							P XY	P B/O	P BΠ	P C/Π

команды $35 = 34 + 1$, стоящий на пересечении 3-го столбца и 5-й строки. Символ F \div будет адресом перехода.

Счетчик адресов используется для управления последовательностью выполнения команд программы. В МК «Электроника Б3-21» команды выполняются по порядку, начиная с адреса 00 до адреса 95. Сброс на нуль счетчика адресов осуществляется нажатием в автоматическом режиме клавиши В/О (возврат-обнуление). Изменение последовательности выполнения команд программы обеспечивают операторы условных и безусловных переходов.

Для ввода программы нужно перейти в режим программирования, нажав клавиши В/О Р РП (режим программирования). В двух последних разрядах индикатора появятся нули 00 — адрес первой команды.

Для перехода в режим автоматических вычислений нажимаем клавиши Р РР (режим работы). Признаком готовности МК к выполнению работы является высвечивание нуля и запятой на индикаторе. При включении микрокалькулятора устанавливается режим автоматических вычислений.

Оперативная и программная память микрокалькулятора построены по «магнитофонному» принципу, т. е. засылка нового числа или команды в регистр автоматически стирает прежнюю информацию, а вызов информации из памяти не изменяет содержимого регистра.

При выключении микрокалькулятора содержимое всех регистров программной памяти автоматически стирается, а содержимое операционных, адресуемых и стековых регистров обнуляется.

Программируемый настольный микрокалькулятор «Электроника МК-46» по своим функциональным возможностям почти не отличается от базовой модели МК «Электроника Б3-21». Он предназначен для выполнения специализированных расчетов, а также для автоматического контроля производственных процессов (в частности, для контроля качества сварочных процессов). К микрокалькулятору могут подключаться внешние устройства. Он может принимать и обрабатывать информацию, поступающую от объекта эксперимента, производить сравнение входных данных с допусковыми значениями, выдавать сигнал отклонения контролируемых параметров от заданных значений, а также выдавать управляющие сигналы на внешние устройства. Результаты обработки информации можно выводить на цифровую печатающую устройство.

Указанный микрокалькулятор имеет программную память емкостью 66 шагов. Программирование производим идентично

МК «Электроника Б3-21», имеется отличие только в символах нескольких операторов:

Б3-21	P PP	P РП
МК-46	P АВТ	P ПРГ

1.5. Микрокалькуляторы «Электроника Б3-34», «Электроника МК-54», «Электроника МК-56», «Электроника МК-61» и «Электроника МК-52»

Передняя панель МК «Электроника Б3-34» представлена на рис. 3. Под световым индикатором расположены: слева выключатель, справа переключатель Р—Г. Порядок включения МК такой же, как и «Б3-21». С помощью переключателя Р—Г можно проводить вычисления с градусной или радианной мерой углов.

Оперативное запоминающее устройство МК «Электроника Б3-34» состоит из четырех регистров стека X, Y, Z, T, регистра X1 восстановления предыдущего результата и четырнадцати адресуемых регистров: 0, 1, 2, ..., 9, A, B, C, D. Для краткости будем их обозначать соответственно RX, RY, ..., R0, R1, ..., RA, ..., RD.

Числа с клавиатуры вводятся в RX. Результаты всех вычислений тоже получаем в RX. Содержимое RX высвечивается на индикаторе.

Для записи числа из RX в какой-либо адресуемый регистр N (N = 0, 1, ..., 9, A, B, C, D) необходимо нажать клавишу П (память) и соответствующую цифровую или буквенную клавишу N. При этом содержимое PN автоматически стирается, число, записываемое в PN, сохраняется в RX. Вызов числа из адресуемого регистра N в RX производится нажатием клавиши ИП (из памяти) и клавиши N. Адресуемые регистры предназначены для хранения исходных данных, промежуточных и окончательных результатов.

Стековые регистры служат для кратковременного хранения и автоматического перемещения информации. Их работа организована по принципу магазинной памяти: последовательно записанная информация считывается в обратном порядке, т. е. последнее записанное число считывается первым. Стек одновходной, числа вводятся или выводятся через регистр X.

Перемещение информации в стековых регистрах X, Y, Z, T и в RX1 осуществляется автоматически с помощью операторов: ↑, ↓ (обозначим XY), F Ø (обозначим F ,),

Таблица 1.2.

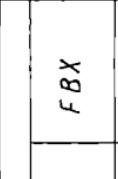
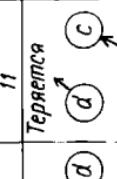
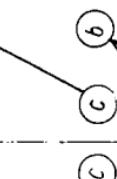
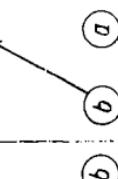
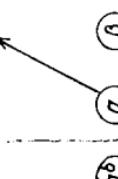
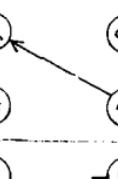
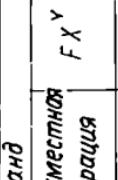
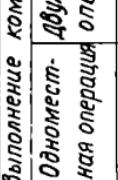
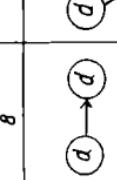
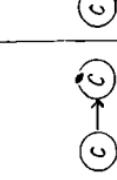
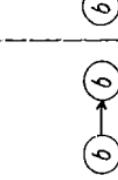
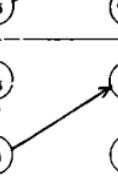
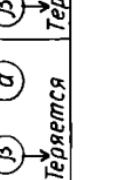
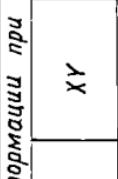
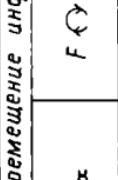
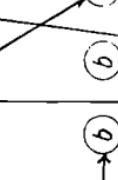
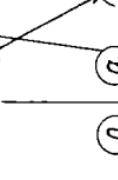
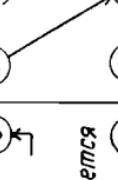
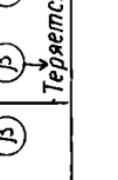
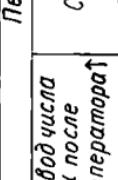
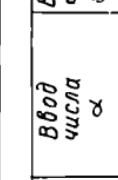
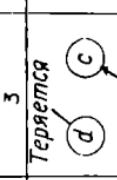
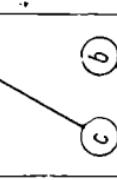
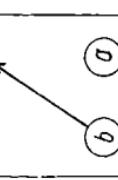
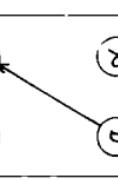
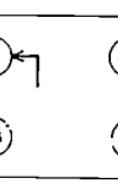
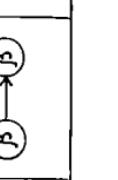
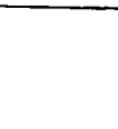
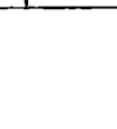
Перемещение информации при выполнение команд									
Последовательность	1	2	3	4	5	6	7	8	9
	Вход числа α	Выход числа α после оператора	C_x	$F \leftrightarrow Y$	$X Y$	Одноместная операция	$F X Y$	$F BX$	
1	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	11
2	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	10
3	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	9
4	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	8
5	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	Теряется 	7
6	Теряется 	6							
7	Теряется 	5							
8	Теряется 	4							
9	Теряется 	3							
10	Теряется 	2							
11	Теряется 	1							



Рис. 3

$F BX$, а также при вводе чисел с клавиатуры, вызове их из адресуемых регистров и при выполнении одноместных и двуместных операций. Схема перемещений показана в табл. 1.2.

При нажатии клавиши \uparrow происходит передача содержимого PX в PY , числа в стеке перемещаются на один регистр «выше» (столбец 2 табл. 1.2). При этом содержимое PX сохраняется, а PT — теряется. В стек можно занести четыре числа a, b, c, d соответственно в регистры T, Z, Y, X , нажимая клавиши: $(a) \uparrow (b) \uparrow (c) \uparrow (d)$.

При вводе числа α с клавиатуры в PX после выполнения какой-либо операции или вызове в PX из адресуемого регистра такое же перемещение информации в стековых регистрах производится автоматически без нажатия клавиши \uparrow (столбец 3 табл. 1.2).

Если в PX вводится число α после оператора \uparrow или производится очистка PX оператором C_x , то содержимое регистров Y, Z, T не изменяется, а в PX заносится соответственно α или 0 (столбцы 4, 5 табл. 1.2).

Оператор $F \Theta$ осуществляет кольцевое передвижение информации в стеке (столбец 6 табл. 1.2), содержимое PX заносится в PT и PX_1 , содержимое PX_1 теряется, во всех остальных регистрах числа сдвигаются «вниз». Оператор XU производит обмен содержимым между PX и PY , содержимое PX заносится также в PX_1 , содержимое PZ и PT не изменяется (столбец 7 табл. 1.2). Регистр восстановления предыдущего результата X_1 сохраняет число, которое находилось в PX до выполнения операции. Для его восстановления следует нажать клавиши $F BX$. Перемещение информации в стеке, вызываемое этим оператором, показано в столбце 11 табл. 1.2.

Во всех рассматриваемых отечественных микрокалькуляторах используется так называемая обратная (инверсная) польская запись математических выражений, предложенная в 1923 году польским математиком Яном Лукасевичем. При такой записи знак операции ставится после набора операндов (чисел) и относится к последнему или двум последним операндам соответственно в одно- и двуместной операции.

При выполнении одноместной операции аргумент a засыпается из PX в регистр восстановления предыдущего результата X_1 , в PX записывается результат операции, содержимое остальных регистров сохраняется (столбец 8 табл. 1.2). При выполнении двуместной операции MK оперирует числами, находящимися в PX и PY , результат операции заносится в PX , предыдущее содержимое которого засыпается в PX_1 ,

содержимое РУ теряется, а в остальных регистрах стека числа перемещаются «вниз» (столбец 9 табл. 1.2).

Отметим некоторую особенность выполнения двуместной операции возвведения в степень x^y . Показатель степени y находится в РУ, основание x — в РХ. После выполнения команды FX^y результат заносится в РХ, содержимое остальных стековых регистров не изменяется (столбец 10 табл. 1.2).

Очистка регистров X, Y, Z, T осуществляется нажатием клавиш C_x, ↑, ↑, ↑. Для очистки РХ1 нажимаем клавишу C_x и любую операционную клавишу, например, +, — и т. д.

Описанный принцип работы стековой памяти позволяет выполнять операции над операндами, хранящимися в четырех стековых регистрах, без дополнительного использования иных регистров памяти, по формулам, включающим три двуместных и достаточное количество одноместных операций, а также выполнять цепочечные вычисления без записи промежуточных результатов. Рассмотрим примеры.

Пример 1. Вычислить $x = \left(\frac{a}{d} + b \right) c$

Вычисления вручную: (c) ↑ (b) ↑ (a) ↑ (d) ÷ + ×

Пусть числа a, b, c, d хранятся соответственно в регистрах 1, 2, 3, 4. Вычисления по программе: ИП3 ИП2 ИП1 ИП4 ÷ + × С/П. Результат получим в регистре X.

Пример 2. Вычислить $x = \frac{\sin(a - \sqrt{bc})}{d^2}$.

Вычисления вручную:

(d) ↑ (a) ↑ (b) ↑ (c) × FV — F sin XY FX² ÷

Вычисления по программе:

ИП4 ИП1 ИП2 ИП3 × FV — F sin XY FX² ÷ С/П

Пример 3. Вычислить $x = A \cos \omega t + B \sin \omega t$.

Вычисления вручную: (A) ↑ (B) ↑ (ω) ↑ (t) × П5 F sin × XY ИП5 F cos × +

Пусть числа A, B, ω, t хранятся соответственно в регистрах 1, 2, 3, 4.

Вычисления по программе: ИП2 ИП3 ИП4 × П5 F sin × ИП1 ИП5 F cos × + С/П

Пример 4. Вычислить определитель 3-го порядка, раскрыв его по элементам первого столбца:

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

Вычисления вручную:

$$(a_{22}) \uparrow (a_{33}) \times (a_{32}) \uparrow (a_{23}) \times - (a_{11}) \times \\ (a_{32}) \uparrow (a_{13}) \times (a_{12}) \uparrow (a_{33}) \times - (a_{21}) \times + \\ (a_{12}) \uparrow (a_{23}) \times (a_{22}) \uparrow (a_{13}) \times - (a_{31}) \times +$$

Разместив элементы определителя в регистры памяти:

$$(a_{11}) \text{ P1}, \quad (a_{12}) \text{ P2}, \quad (a_{13}) \text{ P3}, \\ (a_{21}) \text{ P4}, \quad (a_{22}) \text{ P5}, \quad (a_{23}) \text{ P6}, \\ (a_{31}) \text{ P7}, \quad (a_{32}) \text{ P8}, \quad (a_{33}) \text{ P9},$$

вычислим по программе

$$\begin{array}{ccccccccc} \text{ИП5} & \text{ИП9} & \times & \text{ИП8} & \text{ИП6} & \times & - & \text{ИП1} & \times & \text{ИП8} \\ \text{ИП3} & \times & & \text{ИП2} & \text{ИП9} & \times & - & \text{ИП4} & \times & + & \text{ИП2} \\ \text{ИП6} & \times & & \text{ИП5} & \text{ИП3} & \times & - & \text{ИП7} & \times & + & \text{С/П} \end{array}$$

На панели МК «Электроника Б3-34» имеется 30 клавиш. При нажатии префиксной клавиши F и любой другой клавиши будет выполнен оператор, символ которого указан над клавишей. Сброс действия клавиши F осуществляется с помощью оператора CF. Префиксная клавиша K используется при программировании в косвенной адресации. Кроме описанных выше клавиш ввода чисел и их перемещения в регистрах памяти, на панели имеются: операционные клавиши (+, -, ×, ÷); символы элементарных функций (sin, cos, tg, X², вызов π, arc sin, arc cos, arc tg, 1/X, √, e^x, lg, ln, X^y, 10^x); клавиши режима работы МК (F АВТ — режим автоматической работы, F ПРГ — режим программирования) и клавиши, используемые при программировании (БП — безусловный переход, ПП — переход к подпрограмме или потактовое прохождение программы в режиме автоматической работы; FX < 0, FX = 0, FX ≥ 0, FX ≠ 0 — операторы условных переходов; FL0, FL1, FL2, FL3 — операторы автоматической организации циклов; КНОП — стирание команды при редактировании; В/О — клавиша установления счетчика адресов на нулевой адрес или возврата из подпрограммы в основную программу; ШГ и ШГ — клавиши смещения счетчика адре-

сов на шаг вперед или назад, С/П — клавиша пуска или ОСТАНОВа вычислений).

Переход в режим автоматических вычислений осуществляется нажатием клавиш F АВТ, если микрокалькулятор до этого находился в режиме программирования, или же этот режим устанавливается автоматически после включения МК. В автоматическом режиме осуществляются ввод исходных данных, вычисления по введенной программе, безусловные переходы на любой адрес программы, индикация результата промежуточных и окончательных вычислений, а также проводятся все вычисления с клавиатурой, как на обычном инженерном микрокалькуляторе.

Для перехода в режим программирования следует нажать клавиши F ПРГ. В этом режиме производится ввод программы в память, контроль правильности ввода программы по индикатору и ее редактирование.

Программная память МК состоит из 98 регистров с адресами от 00 до 97 и счетчика адресов. При вводе программы в память каждый оператор автоматически преобразуется в двузначный код, который заносится в очередной регистр памяти, а содержимое счетчика адресов увеличивается на единицу. При этом на индикаторе высвечивается адрес следующей команды и коды трех последних команд. Например, показания индикатора МК «Электроника Б3-34»:

1	3		1	1		6	3		0	9
---	---	--	---	---	--	---	---	--	---	---

означают, что адрес очередной команды равен 09, а 13, 1 \square , 63 — коды трех последовательных команд с адресами 08, 07, 06 соответственно.

Каждому шагу программы, кроме его адреса, отвечает адрес перехода. В МК «Электроника Б3-34» адрес перехода совпадает с адресом команды, на которую требуется перейти. Коды операций и команд МК «Электроника Б3-34» приведены в табл. 1.3, в которой символ строки равен первому знаку кода, а символ столбца — второму. Их удобно использовать для контроля правильности ввода программы и ее редактирования.

Переносной МК «Электроника МК-54» (рис. 4) и настольный МК «Электроника МК-56» (рис. 5) построены на элементной базе модели МК «Электроника Б3-34» и обладают аналогичными с ней функциональными возможностями и входным языком. В этих МК предусмотрена возможность представления углов не только в радианах и градусах, но и в градах

Таблица 1.3

	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	+	-	×	÷	XU	F10 ^X	Fe ^X
2	Fπ	Fν	FX ²	F1/X	FX ^Y	F,	K+
3	K3	K4	K5	K6	K7	K8	K9
4	Π0	Π1	Π2	Π3	Π4	Π5	Π6
5	C/P	BП	B/0	PП	KНОП	K1	K2
6	ИП0	ИП1	ИП2	ИП3	ИП4	ИП5	ИП6
7	KX ≠ 00	KX ≠ 01	KX ≠ 02	KX ≠ 03	KX ≠ 04	KX ≠ 05	KX ≠ 06
8	КБП0	КБП1	КБП2	КБП3	КБП4	КБП5	КБП6
9	KX ≥ 00	KX ≥ 01	KX ≥ 02	KX ≥ 03	KX ≥ 04	KX ≥ 05	KX ≥ 06
—	КПП0	КПП1	КПП2	КПП3	КПП4	КПП5	КПП6
L	KPO	KPI	KP2	KP3	KP4	KP5	KP6
C	KX < 00	KX < 01	KX < 02	KX < 03	KX < 04	KX < 05	KX < 06
Г	КИП0	КИП1	КИП2	КИП3	КИП4	КИП5	КИП6
E	KX = 00	KX = 01	KX = 02	KX = 03	KX = 04	KX = 05	KX = 06

(1 град равен 1/100 части прямого угла). В модели «Электроника МК-54» уменьшены габаритные размеры и масса.

В новых моделях МК «Электроника МК-61» (рис. 6) и «Электроника МК-52» за основу взята система команд МК «Электроника Б3-34», но возможности программирования здесь расширены: добавлен один адресуемый регистр e, число регистров программной памяти равно 105 с адресами от 00 до 104. Для прямых и косвенных команд адреса переходов с 00 по 99 записываются нажатием соответствующих цифровых клавиш. Адреса переходов с 100 по 104 используются только для прямых команд и записываются нажатием клавиши [•] (она соответствует цифре 10) и одной из клавиш

7	8	9	-	L	C	G	E	
7	8	9	,	/-/	BП	C _x	↑	FBX
F lg	F ln	F arcsin	F arccos	F arctg	F sin	F cos	F tg	
K-	K×	K÷	KХУ					
K,	K /-/	KВП	KС _x	K↑				
П7	П8	П9	ПА	ПВ	ПС	ПД	П↑	
FX ≠ 0	FL2	FX > 0	FL3	FL1	FX < 0	FL0	FX = 0	
ИП7	ИП8	ИП9	ИПА	ИПВ	ИПС	ИПД	ИП↑	
KХ ≠ 07	KХ ≠ 08	KХ ≠ 09	KХ ≠ 0A	KХ ≠ 0B	KХ ≠ 0C	KХ ≠ 0D	KХ ≠ 0↑	
КБП7	КБП8	КБП9	КБПА	КБПВ	КБПС	КБПД	КБП↑	
KХ > 07	KХ > 08	KХ > 09	KХ > 0A	KХ > 0B	KХ > 0C	KХ > 0D	KХ > 0↑	
КПП7	КПП8	КПП9	КППА	КППВ	КППС	КППД	КПП↑	
КП7	КП8	КП9	КПА	КПВ	КПС	КПД	КП↑	
KХ < 07	KХ < 08	KХ < 09	KХ < 0A	KХ < 0B	KХ < 0C	KХ < 0D	KХ < 0↑	
КИП7	КИП8	КИП9	КИПА	КИПВ	КИПС	КИПД	КИП↑	
KХ = 07	KХ = 08	KХ = 09	KХ = 0A	KХ = 0B	KХ = 0C	KХ = 0D	KХ = 0↑	

0—4. Многие клавиши этих МК имеют двойную и тройную символику. Цвет символов соответствует цвету префиксных клавиш F и K. Для ввода оператора, символ которого изображен над клавишей, сначала нужно нажать клавишу F или K, а затем клавишу, сверху которой указан этот символ. В этих моделях добавлены операторы: K [X] — выделения целой части числа ($|x|$ равно наибольшему целому числу, не превышающему x); K {X} — выделения дробной части числа; K | X | — определения модуля числа; KЗН — определения знака числа; K пах — выделения из двух чисел, содержащихся в РХ и РУ максимального; КСЧ — генерации случайных чисел в диапазоне от 0 до 1; К О... — перевода вре-



Рис. 4

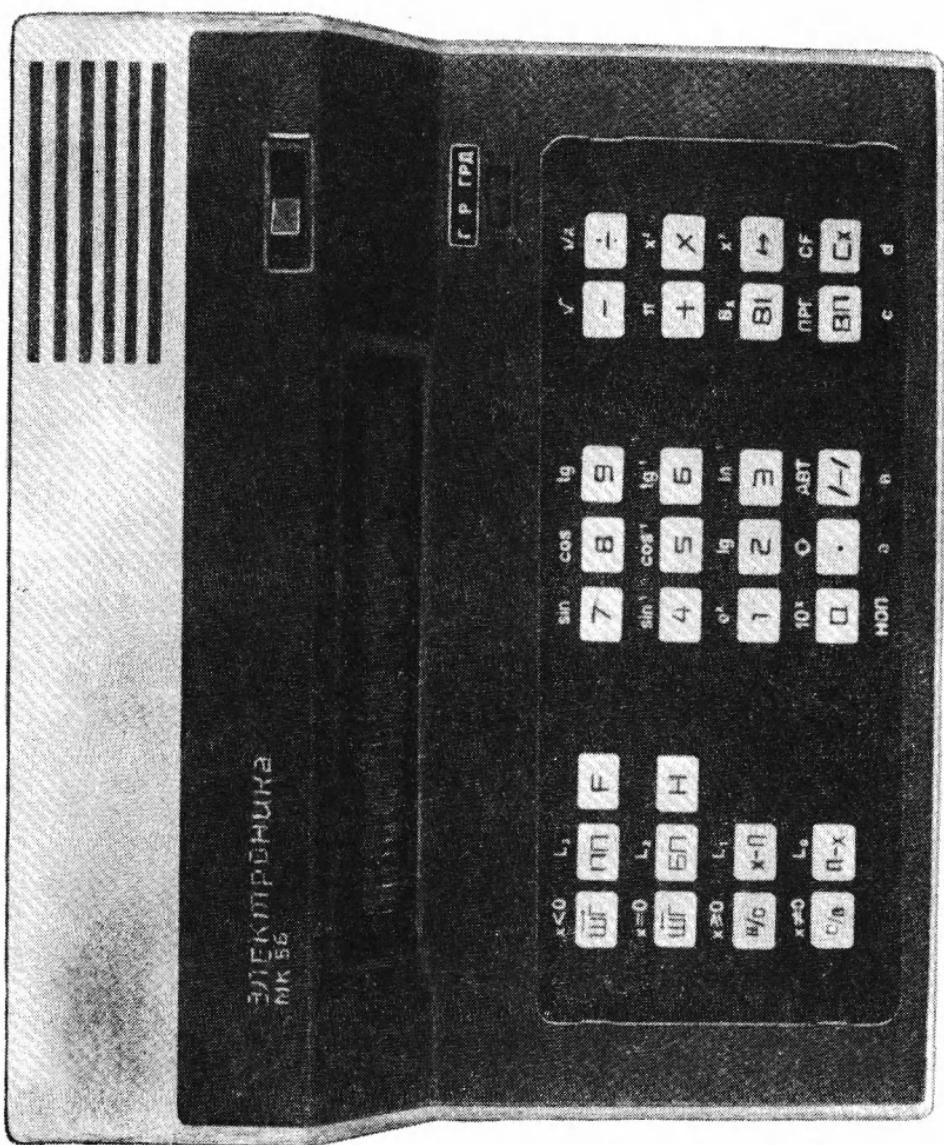


Рис. 5



Рис. 6

менных величин, выраженных в часах, минутах, секундах и долях секунды, в значения часов и долей часа; К $\overset{\leftarrow}{O'}$ — перевода временных величин, выраженных в часах и десятичных долях часа, в значения часов, минут, секунд и долей секунды; К $\overset{\rightarrow}{O'}$ — перевода угловых величин, выраженных в градусах и минутах, в значения градусов и долей градуса; К $\overset{\leftarrow}{\Theta'}$ — перевода угловых величин, выраженных в градусах и долях градуса, в значения градусов, минут и долей минуты; КΛ, К∨, К \oplus , КИНВ — выполнения соответственно логических операций умножения, сложения, сложения по модулю 2, инверсии над числами, находящимися в РХ и РУ.

В отличие от всех предыдущих моделей МК «Электроника МК-52» имеет встроенное полупостоянное запоминающее устройство (ППЗУ), в которое можно записывать программы объемом до 512 шагов, начиная с произвольного адреса. Записанные в ППЗУ программы хранятся в течение длительного времени (до 5000 ч) и не стираются при отключении питания. Их можно считывать в оперативное запоминающее устройство (ОЗУ) МК по мере необходимости и проводить вычисления. В этой модели предусмотрено также подключение модулей (постоянных запоминающих устройств (ПЗУ)) большой емкости с определенным набором программ, записанных заводом-изготовителем.

Следует отметить, что программы, составленные для модели МК «Электроника Б3-34», могут быть реализованы на перечисленных выше моделях микрокалькуляторов. Различия в обозначениях символов некоторых операторов в этих моделях приведены в табл. 1.4.

Таблица 1.4

Б3-34	ИП	П	ХУ	А, В, С, Д	\arcsin	\arccos	\arctg
МК-52, МК-54, МК-56, МК-61	$P \rightarrow X$	$X \rightarrow P$	\leftrightarrow	a, b, c, d, e	\sin^{-1}	\cos^{-1}	\tg^{-1}

Следует также отметить, что в моделях «Электроника МК-52» и «Электроника МК-61» адресуемому регистру е соответствует клавиша В↑. Далее будет установлено существенное различие между командами косвенной адресации с регистром е вида КХ → Пе, КП → Хе и соответствующими им командами КП↑ и КИП↑ в модели «Электроника Б3-34».

1.6. Работа микрокалькулятора в автоматическом режиме

Микрокалькулятор работает в автоматическом режиме и в режиме программирования. Режим автоматической работы устанавливается после включения микрокалькулятора, либо после нажатия клавиш РР («Электроника Б3-21») и F АВТ («Электроника Б3-34»), на индикаторе высвечивается нуль и запятая. В автоматическом режиме МК работает как обычный инженерный калькулятор (выполняется ввод чисел, проводятся вычисления по командам, подаваемым оператором с клавиатуры) либо вычисления проводятся по программе, заранее введенной в программную память.

Вычисления по любой формуле сводятся к последовательному выполнению одноместных и двуместных операций. В одноместной операции вычисляется значение определенной элементарной функции одного операнда — заданного значения аргумента. В двуместной операции (+; —; ×; ÷; X^y) предусматривается действие над двумя числами — операндами.

Для выполнения одноместной операции необходимо:

- 1) ввести в РХ значение аргумента;
- 2) нажать клавиши, соответствующие оператору вычисления нужной элементарной функции.

Результат заносится в РХ, а аргумент — в РХ1.

При вычислении тригонометрических и обратных тригонометрических функций в МК «Электроника Б3-34» устанавливаем переключатель Р—Г в положение радианы или градусы соответственно аргументу. В МК «Электроника Б3-21» аргумент тригонометрических функций должен быть задан в радианах. Переход от градусной меры углов в радианную осуществляется по программе:

(α) ↑ F_л × 1 8 0 ÷

Выполнение двуместной операции осуществляется по схеме:

- 1) вводим первый operand a ;
- 2) пересылаем его в регистр У, нажав клавишу ↑;
- 3) вводим второй operand b ;
- 4) нажимаем клавишу соответствующей операции.

Результат вычислений заносится в РХ, а operand b в РХ1.

При вычислении X^y первым operandом является показатель степени y . В некоммутативных операциях $a — b$ и $a : b$ operand a должен находиться в РУ, а operand b — в РХ.

При вычислении значений элементарных функций необходимо учитывать ограничения, накладываемые на их аргументы областью определения.

1.7. Работа микрокалькулятора в программируемом режиме

В режиме программирования осуществляется ввод программы в память, проверка правильности ввода по индикатору, редактирование и отладка программы.

Ввод программы в память. Если программу необходимо ввести с нулевого адреса, то нажимаем в автоматическом режиме клавишу очистки счетчика адресов В/О, а затем клавиши перехода в режим программирования: Р РП в МК «Электроника Б3-21» и F ПРГ в МК «Электроника Б3-34». На индикаторе справа появится нулевой адрес счетчика 00.

Для занесения программы с произвольного адреса α в автоматическом режиме нажимаем клавишу безусловного перехода БП и клавиши адреса перехода. После перехода в режим программирования на счетчике адресов установится адрес α . Затем последовательно вводим программу, нажимая соответствующие клавиши команд. Визуальный контроль правильности ввода программы осуществляется по индикатору, сверяя коды вводимых команд.

Редактирование программы. Если при вводе программы допущена ошибка, то для ее исправления можно воспользоваться клавишами \vec{W} Г (шаг вперед) и \vec{G} Г (шаг назад) для перемещения счетчика на нужный адрес программы и повторить ввод команды. Парные команды переходов исправляются совместно перемещением на два шага. При большой разнице адресов ошибочной и текущей команд следуют перейти в автоматический режим, нажать клавишу БП и нужный адрес перехода. После перехода в режим программирования на индикаторе появится адрес ошибочной команды, после чего вносится исправление. Для исключения какой-либо команды из программы переходим на ее адрес и вводим оператор КНОП (нет операции), не изменяя дальнейшую программу. Если при вводе программы некоторая команда пропущена, то переходим на ее адрес, вводим нужную команду и все последующие команды.

Отметим, что в режиме программирования операции не выполняются и содержимое регистров оперативной памяти не изменяется.

Отладка программы производится в автоматическом режиме по просчитанному вручную контрольному примеру. Ее

проводят для устранения ошибок, допущенных при составлении программы. Набираем на клавиатуре исходные данные для контрольного примера и заносим их в регистры памяти. Нажимаем клавишу В/О, если программа начинается с адреса 00, или БП, а затем клавиши требуемого адреса, если программа начинается с произвольного адреса. Последовательно нажимаем клавишу ПП, при этом происходит пошаговое выполнение программы. Проверяем работу каждого шага программы, сверяя показания индикатора с теми значениями, которые должны получиться. При пошаговом просмотре программы следует учитывать, что выполнение команды перехода и установка адреса перехода осуществляются за один шаг программы.

В настоящей книге по каждой лабораторной работе приведены отлаженные программы. Поэтому следует проверить безошибочность ввода программы, посчитав контрольный пример и сверив ответ с приведенным в книге. Если ответ не совпадает, то повторить ввод программы, если совпадет, то можно выполнять свой вариант задания.

1.8. Составление программы

Решение различных задач на ЭВМ любого типа, в том числе и на программируемых микрокалькуляторах, включает в себя следующие этапы:

- 1) постановка задачи и ее математическое описание;
- 2) выбор численного метода решения задачи;
- 3) составление алгоритма вычислений (или его блок-схемы);
- 4) составление программы, ее редактирование и отладка;
- 5) вычисления по программе;
- 6) анализ полученных результатов.

Понятие алгоритма является основным понятием математики. Алгоритм — точно определенное правило действий (программа), для которого задано указание, как и в какой последовательности это правило применять к исходным данным задачи, чтобы получить ее решение. Алгоритмы могут представляться в виде формульно-словесного описания или блок-схемы.

Блок-схема алгоритма графически изображает вычислительный процесс, который должен быть реализован соответствующей программой на ЭВМ. При этом отдельные операторы или группы операторов называют блоками. Различают такие блоки: арифметические, логические, ввода и вывода

данных, блоки начала и конца алгоритма. Арифметические блоки содержат арифметические операторы и изображаются в виде прямоугольников, внутри которых записывается формула вычислений. Логические блоки предусматривают проверку некоторых условий. Их изображают ромбами, внутри которых записывают проверяемое условие. Блоки ввода и вывода изображаются параллелограммами. Блоки начала и конца алгоритма изображаются овалами. Блоки соединяют между собой в определенном порядке стрелками, которые указывают последовательность вычислений. Из каждого арифметического блока выходит одна стрелка. Из логического блока выходят две стрелки. Одна стрелка со словом «Да» указывает на блок, который будет продолжать вычислительный процесс в случае выполнения проверяемого условия. Другая стрелка со словом «Нет» указывает на блок, по которому будут проводиться вычисления, если проверяемое условие не выполняется.

Алгоритм и блок-схема делают более понятной и наглядной схему вычислительного процесса данного метода. Для того чтобы метод был «понятен» машине, нужно составить его программу на соответствующем машинном языке.

Программа является описанием алгоритма решения задачи, заданным на языке вычислительной машины. Описание программы в программируемых микрокалькуляторах символьно-кодовое, т. е. каждой операции соответствует определенный символ, который машина автоматически преобразует в код. Символы операторов в основном совпадают с обозначениями клавиш, которые необходимо нажимать при вводе программы. Их назначение указывается в инструкции к микрокалькулятору. Код каждого оператора приводится в таблице кодов, с помощью которой удобно сверять правильность ввода программы и проводить ее отладку. Символы операторов (команд) в программе соединены в последовательность по определенным синтаксическим правилам. Каждая команда в программе занимает строго определенное место, называемое ее *адресом*.

Составленную и введенную в память микрокалькулятора программу можно использовать многократно, изменяя только исходные данные. Числовые данные можно записывать в программную память, но в случае многозначных чисел это делать неэкономно, так как каждая цифра числа, десятичная запятая, знак числа, знак порядка занимают по одной ячейке программы памяти. Исходные данные предварительно заносят в регистры памяти, предусматривая в программе команды вызова их содержимого в процессе вычислений.

Для прекращения вычислений и индикации результатов в программе обязательно записываем команду автоматического ОСТАНОВА С/П.

Самыми простыми являются *линейные программы*, в которых команды выполняются последовательно одна за другой в том порядке, в каком они записаны, без повторений. В таких программах весь ход вычислений предопределяется заранее однозначно.

Если алгоритм метода предусматривает разделение вычислительного процесса на несколько ветвей в зависимости от конкретных значений результатов выполнения предыдущих команд, то такие алгоритмы и их программы называют *разветвляющимися*. Они позволяют описывать достаточно сложные вычислительные процессы с помощью логических операторов условных и безусловных переходов.

Операторы безусловного и условного переходов изменяют последовательный порядок выполнения программы и обеспечивают переход на любой ее адрес. Оператор безусловного перехода обозначается символом БП, за ним указывается адрес перехода. Команды условных переходов предусматривают проверку содержимого регистра X на одно из следующих условий: $X < 0$, $X = 0$, $X \geq 0$, $X \neq 0$. Если условие не выполняется, то следующей по программе будет выполняться команда перехода по указанному адресу. Если условие выполняется, то следующей будет исполнена команда, записанная в программе после адреса перехода, при этом адрес перехода не воспринимается.

В некоторых задачах приходится повторять одни и те же вычисления в нескольких местах программы. Тогда удобнее эти вычисления выделить в виде *подпрограммы* и обращаться к ней в любой части основной программы по мере необходимости. Подпрограммы записывают, как правило, в конце основной программы. Таких подпрограмм может быть несколько. В частности, подпрограмма может содержать обращение к другой подпрограмме, при этом глубина обращений не должна превышать пяти. Обращение к подпрограмме в микрокалькуляторах производится с помощью команды ПП — перехода к подпрограмме. После этой команды указывается адрес перехода. Последним оператором каждой подпрограммы является команда В/О — возврата в основную программу, по которой возобновляются вычисления по основной программе, начиная с номера команды, следующей после адреса перехода к подпрограмме.

Если в программе некоторая группа операций повторяется несколько раз, пока не будет выполняться определенное усло-

вие, то такие программы называют *циклическими*. Группу повторяющихся команд называют *циклом*. Количество повторений цикла контролирует логический оператор, который осуществляет передачу управления при выходе из цикла. Циклы, в которых количество повторений задается наперед, называют *арифметическими*. Примером таких циклических программ может служить программа табулирования функции $f(x)$ на отрезке $[a; b]$ с шагом h . При этом вычисление значений функции повторяется $n + 1$ раз, где $n = (b - a) / h$, каждый раз при новом значении аргумента $x_i = a + ih$ ($i = 0, 1, \dots, n$). Число, равное количеству повторений цикла, засыпается предварительно в один из адресуемых регистров, называемый *счетчиком цикла*. При численном решении задач итерационными методами используются *итерационные циклы*. Число повторений в таких циклах заранее неизвестно, а выход из цикла осуществляется при достижении наперед заданной точности.

В МК «Электроника Б3-34» и построенных на его основе микрокалькуляторах имеются специальные операторы автоматической организации циклов: FL0, FL1, FL2 и FL3. При нажатии клавиши FLN ($N = 0, 1, 2, 3$) происходит обращение к регистру N, из его содержимого вычитается единица и производится проверка результата по условию X = 0. Если получившаяся разность не равна нулю, то она заносится в PN, а управление передается по адресу, записанному в программе после оператора цикла; если разность равна нулю, то выполняется команда, записанная после адреса перехода, а нуль в PN уже не записывается.

Прямая и косвенная адресация. Если при программировании адрес регистра памяти или адрес перехода указывается непосредственно в программе, то такую адресацию называют *прямой*. Она характерна для МК «Электроника Б3-21». В МК «Электроника Б3-34», кроме прямой, предусмотрена также *косвенная адресация*. Суть ее состоит в том, что вместо указания адреса перехода или регистра памяти в команде указывается место в памяти (регистр адресации), где расположен нужный адрес.

Оператор косвенной адресации занимает один шаг программы и обозначается с помощью клавиши K, клавиши оператора перехода или клавиши оператора обращения к регистру памяти N. Например, команды косвенного обращения к подпрограмме, косвенного безусловного и условных переходов по модифицированному адресу, указанному в регистре адресации N, реализуются клавишами: КПП N, КБП N, KX < 0N, KX = 0N, KX ≥ 0N, KX ≠ 0N. При каждом

выполнении такой команды производится модификация адреса, хранящегося в регистре адресации N, и осуществляется переход к выполнению команды, записанной по новому (модифицированному) адресу. Если логическое условие в условных переходах выполняется, то модификация адреса не происходит, и вычисления, как обычно, проводятся по команде, следующей за оператором перехода.

Команды косвенной записи и косвенного вызова реализуются клавишами КП N, КИП N. При этом производится модификация содержимого регистра адресации N и запись информации в регистр (или вызов из регистра) с модифицированным номером. Регистрам А, В, С, Д соответствуют номера 10, 11, 12, 13.

Модификация адреса в зависимости от номера N регистра адресации осуществляется следующим образом: если $N = 0, 1, 2, 3$, то из содержимого регистра вычитается единица, если $N = 4, 5, 6$, то к его содержимому прибавляется единица, для всех остальных номеров содержимое регистра не изменяется.

Отметим некоторую особенность двух команд косвенной адресации КП \uparrow и КИП \uparrow . Команда КП \uparrow , как и команда КПО, засыпает содержимое РХ в адресуемый регистр, номер которого хранится в Р0. Но в отличие от команды КПО содержимое Р0 не уменьшается на единицу, т. е. не модифицируется. Аналогично команда КИП \uparrow производит вызов содержимого из адресуемого регистра, номер которого хранится в Р0, без модификации адреса в отличие от команды КИПО.

1.9. Ввод исходных данных и пуск программы

Исходные данные, необходимые для решения задачи по введенной в память МК программе, вводятся в автоматическом режиме. Каждое вводимое число набирается на клавиатуре, затем нажимаются клавиши П и N — номер нужного адресуемого регистра. Если количество вводимых чисел сравнительно велико, то такой порядок ввода требует определенного напряжения от оператора. Значительно проще предусмотреть ввод чисел в нужные регистры в самой программе, если это позволяет программная память. Оператору тогда нужно набирать на клавиатуре числа и нажимать после каждого клавишу С/П. Например, ввод чисел a, b, c, d соответственно в регистры А, В, С, Д можно осуществить следующими командами программы: ПА С/П ПВ С/П ПС С/П ПД

Ввод чисел можно осуществлять с помощью клавиши ПП, не предусматривая ОСТАНОВЫ в программе. Программа предыдущего примера содержит команды ПА ПВ ПС ПД, а оператор нажимает клавиши (a) ПП (b) ПП (c) ПП (d) ГП. Команда ПП обеспечивает выполнение только одной команды программы.

Для ввода массива чисел в идущие подряд адресуемые регистры можно использовать команды косвенной засылки КПН. Выбор регистра адресации N обеспечивает запись чисел в регистры с возрастающими или убывающими номерами. Например, если выбрать Р4 в качестве регистра адресации, а Р0 в качестве счетчика цикла, то ввод девяти чисел x_1, x_2, \dots, x_9 соответственно в регистры 5, 6, ..., 9, А, В, С, Д можно осуществить с помощью такого фрагмента программы:

4 П4 9 П0 С/П КП4 FL0 04 ...

Оператор последовательно нажимает клавиши В/О, С/П, (x_1), С/П, (x_2), С/П, ..., (x_9), С/П. По команде КП4 содержимое Р4 (равное вначале 4) модифицируется, т. е. увеличивается на единицу, и x_1 заносится в Р5. По команде FL0 из содержимого Р0 (равного вначале 9) вычитается единица, полученная разность сравнивается с нулем и посыпается в Р0, а управление передается на команду с адресом 04, т. е. происходит ОСТАНОВ. Следующее число x_2 заносится в Р6, а содержимое Р0 становится равным 7. После ввода последнего числа осуществляется выход из цикла ввода на выполнение основной программы.

Если выбрать Р0 в качестве регистра адресации и счетчика циклов, то ввод n чисел в регистры 1, 2, 3, ..., n ($n \leq 13$) в последовательности убывания их номеров удобно производить с помощью такого цикла ввода

Пn (n - 1) П0 С/П КП↑ FL0

После FL0 ставим адрес перехода на команду С/П. В этом случае ввод чисел x_1, x_2, \dots, x_n соответственно в регистры $n, n - 1, \dots, 2, 1$ осуществляется последовательным нажатием клавиш (x_1) В/О С/П (x_2) С/П ... (x_n) С/П.

После ввода исходных данных осуществляем вычисления по программе в автоматическом режиме. Нажимаем клавишу В/О, если выполнение программы начинается с нулевого адреса, или клавишу БП и клавиши адреса перехода на α , если программа начинается с адреса α . Затем нажимаем клавишу пуска программы С/П. Слабое мелькание на индикаторе свидетельствует об автоматическом вычислении по программе. В некоторых случаях в программе могут быть преду-

смотрены ОСТАНОВЫ в процессе вычислений по программе для индикации промежуточных результатов или же ввода данных. Для дальнейшего прохождения программы нажимаем только клавишу пуска С/П. Если в программе есть ошибки, то возможны зациклования. В таких случаях следует нажать клавишу останова С/П и исправить ошибку.

Для проведения многократных вычислений по отлаженной программе заносим в регистры памяти новые исходные данные, восстанавливаем счетчики циклов, осуществляя очистку требуемых регистров памяти, если это не предусмотрено в самой программе, и повторяем пуск программы.

1.10. Запись программы

Программа численного метода записывается в виде таблицы, в которой указаны: адрес команды, символ оператора, код операции и комментарий по содержанию операций.

Такая запись позволяет контролировать правильность ввода программы в память МК по кодам операций, а также делает более наглядным алгоритм вычислений.

Применяется также более компактная запись программы в строку с фиксированным числом операторов. При такой записи адреса команд и коды операторов не записываются. Программа на входном языке МК «Электроника Б3-21» содержит 12, а на входном языке МК «Электроника Б3-34» — 10 операторов в строке. В зависимости от числа команд в программе последняя строка может быть неполной. Такая запись позволяет легко определить адрес любой команды, а также адрес перехода.

В инструкции по работе с программой указывается, как перейти к решению нового варианта задания. Если для решения задачи требуется подпрограмма вычисления значений конкретной функции, то она составляется и записывается в основную программу.

В качестве примера приведем описание алгоритма, блок-схемы и программы для приближенного вычисления квадратного корня из положительного числа a с помощью рекуррентной формулы:

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right), \quad n = 0, 1, 2, \dots \quad (1.1)$$

Формула эта была известна еще древнегреческому математику Герону Александрийскому (I век до н. э.). При любом начальном приближении $x_0 > 0$ итерационный процесс (1.1) сходится, при этом $x_n \geq \sqrt{a}$ ($n = 1, 2, 3, \dots$). Вычисления можно

прекратить, если для любой заданной точности $\epsilon > 0$ выполняется неравенство

$$x_n - \sqrt{a} < \epsilon \text{ или } x_n - \sqrt{a} = \frac{x_n^2 - a}{x_n + \sqrt{a}} < x_n - \frac{a}{x_n} < \epsilon. \quad (1.2)$$

Словесное описание алгоритма.

1. Ввести a , x_0 , ϵ : (а) П1, (x_0) П2, (ϵ) П3.
2. Положить $x = x_0$.
3. Вычислить $y = \frac{a}{x}$.
4. Вычислить $z = |x - y|$.
5. Если $z - \epsilon < 0$, то перейти к п. 7, иначе к п. 6.
6. Вычислить: $x = \frac{1}{2}(x + y)$ и перейти к п. 3.
7. Положить $\sqrt{a} = x$. Конец.

Блок-схема алгоритма представлена на рис. 7.

Запишем программу алгоритма на языке МК «Электроника Б3-34»:

```
ИП1 ИП2 ÷ П0 ИП2 — FX<0 09 /—/ ИП3
— FX<0 15 ИП2 С/П ИП2 ИП0 + 2 ÷
П2 БП 00
```

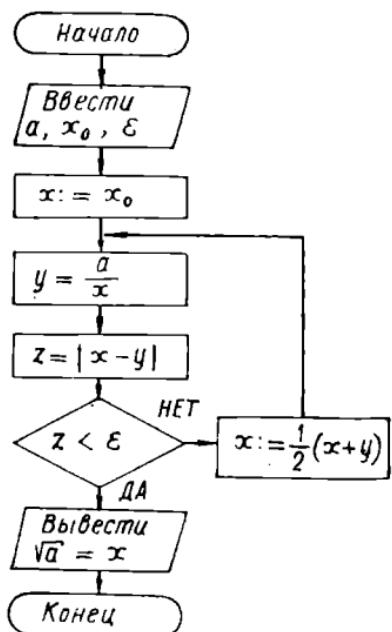


Рис. 7

Инструкция. В режиме программирования (нажав клавиши F ПРГ) ввести программу. В режиме автоматической работы (нажав клавиши F АВТ) ввести исходные данные (см. п. 1). Осуществить пуск программы, нажав клавиши В/О, С/П. После ОСТАНОВа на индикаторе высвечивается приближенное значение \sqrt{a} .

Методика составления программ и вычислений на микрокалькуляторах содержится в литературе [3, 16, 17, 29—31, 36, 37].

Глава 2. ЭЛЕМЕНТЫ ТЕОРИИ ПОГРЕШНОСТЕЙ

2.1. Классификация погрешностей

При решении инженерных задач на любых вычислительных устройствах расчеты, как правило, производятся над приближенными исходными данными. В таких случаях необходимо уметь грамотно организовать вычисления, знать, какие бывают ошибки, правильно записывать приближенные исходные данные, оценивать ошибку результата по известным ошибкам компонент, определять оптимальное количество знаков в исходных данных, чтобы результат вычислений имел наперед заданную точность, выбирать наиболее рациональный порядок вычислений, а также алгоритм вычислений, устойчивый к ошибкам округлений.

Не имея достаточных навыков практических вычислений, можно при решении задачи получить результат, не имеющий ничего общего с действительным решением задачи.

Ознакомимся с основными понятиями теории погрешностей.

Основные источники погрешности приближенного решения прикладных задач следующие.

Погрешность математической модели. Она связана с физическими допущениями, не контролируется в процессе численного решения задачи и может уменьшиться только за счет более точного описания физической задачи.

Погрешность исходных данных. Значения параметров, входящих в математическое описание задачи, измеряются экспериментально с некоторой погрешностью.

Погрешность математической модели и исходных данных называют *неустранимой*. Ее необходимо учитывать при выборе метода решения задачи.

Погрешность приближенного метода. При численном решении задачи точный оператор, в котором количество чисел или операций превышает допустимые границы, заменяется приближенным, требующим конечного количества операций. Например, заменяют интеграл суммой, функцию — многочленом или строят бесконечный итерационный процесс и обрывают его после конечного числа итераций. Эту погрешность будем исследовать при рассмотрении конкретных численных методов.

Вычислительная погрешность, возникающая в результате вынужденного округления чисел, например, конечного числа разрядов ЭВМ.

Если решение некоторой задачи непрерывно зависит от входных данных, т. е. малому изменению входных данных

соответствует малое изменение решения, то задача называется устойчивой по входным данным. В устойчивом вычислительном алгоритме ошибки округления не накапливаются.

Рассмотрим общие правила действий над приближенными числами и оценки получающихся при этом погрешностей.

2.2. Абсолютная и относительная погрешности

Точность приближенного числа характеризуется понятиями абсолютной и относительной погрешности.

Абсолютной погрешностью приближенного числа a называется величина

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

где A — точное значение некоторой величины, a — одно из его приближенных значений. Абсолютная погрешность Δ представляет только теоретический интерес, поскольку точное значение A не известно. Поэтому на практике чаще пользуются предельной абсолютной погрешностью Δ_a приближенного числа a , равной по возможности наименьшему числу, для которого выполняется неравенство

$$\Delta = |A - a| \leq \Delta_a.$$

Значения a и Δ_a позволяют указать интервал, содержащий точное значение A :

$$a - \Delta_a \leq A \leq a + \Delta_a.$$

Чаще используется более компактная запись

$$A = a \pm \Delta_a.$$

Основной характеристикой точности приближенного числа является его относительная погрешность

$$\delta = \frac{\Delta}{|A|}.$$

Поскольку число A неизвестно, то, как правило, полагают

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

Аналогично из неравенства $\delta \leq \delta_a$ определяют предельную относительную погрешность числа a , полагая

$$\delta_a = \frac{\Delta_a}{|a|}.$$

Величина δ характеризует качество приближения. Это безразмерная величина, обычно ее выражают в процентах.

2.3. Запись приближенных чисел. Правила округления

Записывать приближенное число в виде $a \pm \Delta_a$ неудобно. Поэтому в вычислительной практике часто прибегают к различным приемам, позволяющим только по записи самого приближенного числа a судить о его погрешности.

Пусть приближенное число a представлено в виде конечной десятичной дроби:

$$a = \alpha_m \cdot 10^m + \alpha_{m-1} \cdot 10^{m-1} + \cdots + \alpha_{m-n+1} \times \\ \times 10^{m-n+1} (\alpha_m \neq 0),$$

где α_i — цифры числа a ($\alpha_i = 0, \dots, 9$).

Значащими цифрами числа называют все цифры в записи числа, начиная с первой ненулевой слева.

Например, у чисел $a = 0,0\overline{5}03$, $b = 0,0\overline{063}0500$ значащими цифрами являются подчеркнутые цифры.

Значащую цифру называют *верной*, если абсолютная погрешность числа не превосходит единицы разряда, соответствующего этой цифре. В противном случае цифра считается *сомнительной*.

Например, в записи $a = 0,0\overline{5}03$, $\Delta_a = 0,00002$; $b = 0,0\overline{063}0500$, $\Delta_b = 0,000008$ подчеркнутые цифры будут *верными*.

Если исходное число имеет несколько сомнительных цифр, его следует предварительно округлить. Приходится также округлять точные числа, содержащие слишком много или бесконечное количество цифр.

Чтобы округлить число до n значащих цифр, отбрасывают все цифры его, стоящие справа от n -й значащей цифры, или, если это необходимо для сохранения разрядов, заменяют их нулями. При этом руководствуются следующим правилом: последняя сохраняемая цифра не изменяется, если первая отбрасываемая или заменяемая цифра меньше 5, и последняя сохраняемая цифра увеличивается на единицу, если первая отбрасываемая или заменяемая цифра больше или равна 5. При отбрасывании или замене только одной цифры 5 последняя сохраняемая цифра не изменяется, если она четная, и увеличивается на единицу, если она нечетная.

Если приближенные числа не носят окончательного характера и с ними предполагается производить еще вычисления, то следует сохранять одну или две сомнительные цифры.

По правилу, предложенному акад. А. Н. Крыловым, приближенное число надо писать так, чтобы в нем все значащие

цифры, кроме последней, были верными и лишь последние цифра была бы сомнительной, притом величина абсолютной погрешности не должна превышать половины единицы последнего оставленного разряда.

Величина относительной погрешности приближенного числа связана с количеством n верных знаков этого числа и его первой значащей цифрой α_m соотношением:

$$\delta_a \leq \frac{1}{2\alpha_m \cdot 10^{n-1}}.$$

Эта формула позволяет грубо оценить количество n верных знаков приближенного числа, если известны его относительная погрешность δ_a и α_m .

2.4. Погрешности функций

Пусть задана некоторая функция $y = f(x_1, x_2, \dots, x_n)$ от n аргументов x_1, x_2, \dots, x_n , причем значения каждого из ее аргументов x_i определены с некоторыми абсолютными погрешностями $\Delta x_i, i = 1, 2, \dots, n$. Требуется найти погрешность данной функции. Предположим, что функция непрерывно дифференцируема в области D . Обозначим величину абсолютной погрешности функции через Δy , т. е.

$$\Delta y = |f(x_1 + \Delta x_1, x_2 + \Delta x_2, \dots, x_n + \Delta x_n) - f(x_1, x_2, \dots, x_n)|.$$

Предполагая, что величины Δx_i достаточно малы, абсолютную погрешность функции будем считать приближенно равной ее дифференциальному

$$\Delta y \approx |df(x_1, x_2, \dots, x_n)| = \left| \sum_{i=1}^n \frac{\partial f}{\partial x_i} \Delta x_i \right| \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \Delta x_i. \quad (2.1)$$

Из полученного неравенства определим верхнюю границу для Δy :

$$\Delta_y = \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \cdot \Delta x_i, \quad (2.2)$$

где через $\Delta x_i (i = 1, 2, \dots, n)$ обозначены предельные абсолютные погрешности аргументов x_i и через Δ_y — предельная абсолютная погрешность функции.

Оценка для относительной погрешности функции получается путем деления обеих частей неравенства (2.1) на $|y|$:

$$\delta \leq \sum_{i=1}^n \left| \frac{1}{y} \right| \left| \frac{\partial f}{\partial x_i} \right| \Delta x_i = \sum_{i=1}^n \left| \frac{\partial}{\partial x_i} \ln y \right| \cdot \Delta x_i.$$

Поэтому предельную относительную погрешность функции y можно принять равной

$$\delta_y = \sum_{i=1}^n \left| \frac{\partial}{\partial x_i} \ln y \right| \Delta x_i. \quad (2.3)$$

Применим общие формулы (2.2) и (2.3) к некоторым частным случаям.

1. Погрешность алгебраической суммы.

Пусть $y = x_1 + x_2 + \dots + x_n$. По формулам (2.2) и (2.3) имеем:

$$\Delta_y = \Delta_{x_1} + \Delta_{x_2} + \dots + \Delta_{x_n}; \quad (2.4)$$

$$\delta_y \leq \max(\delta_{x_1}, \delta_{x_2}, \dots, \delta_{x_n}). \quad (2.5)$$

Для разности $y = x_1 - x_2$ получим аналогично

$$\Delta_y = \Delta_{x_1} + \Delta_{x_2}. \quad (2.6)$$

При этом заметим, что если числа x_1 и x_2 мало отличаются друг от друга, то предельная относительная погрешность разности может стать достаточно большой, хотя относительные погрешности δ_{x_1} и δ_{x_2} малы. Здесь происходит потеря точности. Поэтому при практических вычислениях формулы, в которые входит разность близких чисел, следует преобразовать так, чтобы это затруднение было устранено.

2. Погрешность произведения и частного.

Пусть $y = x_1 x_2 \dots x_n$, причем числа x_i ($i = 1, 2, \dots, n$) положительны. Отсюда $\ln y = \sum_{i=1}^n \ln x_i$. Согласно формуле (2.3)

$$\delta_y = \delta_{x_1} + \delta_{x_2} + \dots + \delta_{x_n}. \quad (2.7)$$

Если $y = \frac{x_1}{x_2}$, то аналогично

$$\delta_y = \delta_{x_1} + \delta_{x_2}. \quad (2.8)$$

3. Погрешности степени и корня.

Пусть $y = x^n$. Тогда $\ln y = n \ln x$,

$$\delta_y = n \delta_x. \quad (2.9)$$

Если $y = \sqrt[n]{x}$, то $\ln y = \frac{1}{n} \ln x$,

$$\delta_y = \frac{1}{n} \delta_x. \quad (2.10)$$

На основании полученных формул (2.4)–(2.10) погрешность результатов действий над приближенными числами выражается через погрешности компонент по следующим правилам:

1. Предельная абсолютная погрешность алгебраической суммы равна сумме предельных абсолютных погрешностей слагаемых.

2. Предельная относительная погрешность суммы не превышает наибольшей из предельных относительных погрешностей слагаемых.

3. Предельная относительная погрешность произведения и частного равна сумме предельных относительных погрешностей компонент.

4. Предельная относительная погрешность n -й степени приближенного числа в n раз больше предельной относительной погрешности данного числа (n может быть как целое, так и дробное).

В практических расчетах с большим числом операций погрешность каждой операции обычно не определяется. Вместо этого на практике используют простые правила округления результатов действий над приближенными числами в зависимости от числа верных значащих цифр или десятичных знаков в исходных данных. Отметим, что *десятичными знаками* числа называются все его верные цифры, стоящие справа после запятой. Сформулируем эти правила.

1. При сложении и вычитании приближенных чисел в результате сохраняем столько десятичных знаков, сколько их в приближенном данном с наименьшим числом десятичных знаков.

2. При умножении и делении в результате сохраняем столько значащих цифр, сколько их имеет приближенное данное с наименьшим числом значащих цифр.

3. При возведении в квадрат или куб и при извлечении квадратного или кубического корня в результате оставляем столько значащих цифр, сколько их имеет данное число.

4. Во всех промежуточных результатах необходимо брать на одну цифру больше, чем рекомендуется в предыдущих правилах. В окончательном результате эта запасная цифра отбрасывается.

5. Если некоторые данные имеют больше десятичных знаков (в правиле 1) или больше значащих цифр (в правилах 2 и 3), чем другие, то их предварительно следует округлить, сохранив лишь одну лишнюю цифру.

6. Если исходные данные можно брать с произвольной точностью, то для получения результата с k верными цифрами

исходные данные берем с таким числом верных цифр, которое дает согласно правилам 1—3 $k + 1$ цифру в результате.

При соблюдении перечисленных правил окончательный результат почти всегда будет иметь все верные цифры.

Рассмотрим задачу: определить допустимую погрешность аргументов по заданной предельной погрешности функции. Эта задача решается однозначно только для функции одной переменной $y = f(x)$. Если $f'(x)$ — дифференцируема и $f'(x) \neq 0$, то из (2.2) при $n = 1$ получим

$$\Delta_x = \frac{1}{|f'(x)|} \Delta_y. \quad (2.11)$$

В случае функции нескольких переменных следует вводить некоторые дополнительные ограничения. Можно применить, например, принцип равных влияний, т. е. считать, что в формуле (2.2) все слагаемые равны. Тогда предельные абсолютные погрешности аргументов могут быть вычислены следующим образом:

$$\Delta_{x_i} = \frac{\Delta_y}{n \left| \frac{\partial f}{\partial x_i} \right|}, \quad i = 1, 2, \dots, n. \quad (2.12)$$

Если предположить, что предельные абсолютные погрешности всех аргументов равны, т. е. $\Delta_{x_1} = \Delta_{x_2} = \dots = \Delta_{x_n}$, то

$$\Delta_{x_i} = \frac{\Delta_y}{\sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right|}. \quad (2.13)$$

2.5. Погрешности вычислений на микрокалькуляторах

Одной из основных особенностей вычислений на микрокалькуляторах является конечное число разрядов индикатора. Поэтому, если число разрядов в исходных данных или промежуточных результатах превышает разрядность индикатора, то последние разряды такого числа отбрасываются. Если количество вычислений велико, то будет накапливаться погрешность округлений и значительно понижаться точность полученного результата.

Потеря точности может происходить при сложении, вычитании, умножении чисел разных порядков, например $285340 + + 0,0095 = 285340,01$, при делении чисел, если порядок делимого значительно меньше порядка делителя, и вычисления проводятся в режиме с фиксированной запятой. В некото-

рых случаях результаты промежуточных вычислений могут попадать в область машинного пулья — на индикаторе не появляется ни одной значащей цифры, или в область бесконечности — происходит переполнение разрядной сетки. Этого частично можно избежать, если представлять числа в стандартной форме с плавающей запятой, что осуществляется автоматически во всех программируемых микрокалькуляторах. Кроме того, удобно применять нормировку исходных данных и коэффициентов вычислительных формул, умножив их на постоянный множитель, уравнивающий порядки.

Значения элементарных функций вычисляются по встроенным микропрограммам тоже с некоторой погрешностью. Ее обычно указывают в инструкции. В микрокалькуляторах используются разложения функций в бесконечные степенные ряды, представления бесконечными цепными дробями или итерационные процессы, которые приходится обрывать на некотором конечном шаге, что также оказывает влияние на окончательный результат.

При вычислениях над приближенными числами погрешность результата зависит от порядка выполнения вычислений.

Пример.

Вычислим выражение $(1 \times 6) : 6$ с точностью до двух знаков после запятой: $1,00 \times 6,00 = 6,00$; $6,00 : 6,00 = 1,00$. Поскольку $(1 \times 6) : 6 = (1 : 6) \times 6$, то $1,00 : 6,00 = 0,17$; $0,17 \times 6,00 = 1,02$. Вычислим, например, сумму

$$100 + \underbrace{0,1 + 0,1 + \dots + 0,1}_{2000 \text{ раз}}$$

округляя результаты до трех значащих цифр: $100 + 0,1 = 100,1 \approx 100$ и т. д. Окончательный результат будет 100. Если вычисления вести в обратном порядке, получим сумму: $0,1 + 0,1 = 0,2$; $0,2 + 0,1 = 0,3$; ...; $99,9 + 0,1 = 100$; $100 + 0,1 = 100$; ...; $100 + 100 = 200$ — окончательный результат. При сложении группами: сложим 1000 раз слагаемые 0,1, затем еще 1000 раз слагаемые 0,1, затем прибавим 100, получим правильный результат — 300.

В некоторых задачах алгоритм вычислений может оказаться неустойчивым к погрешностям округлений. Например, если в системе

$$\begin{cases} 10,1x + 9,9y = 20,0, \\ 9,9x + 10,1y = 20,0, \end{cases}$$

имеющей решение $(1; 1)$, изменить на 0,5 % правые части

$$\begin{cases} 10,1x + 9,9y = 20,1, \\ 9,9x + 10,1y = 19,9, \end{cases}$$

то решение $(1,5; 0,5)$ отличается от первоначального на 50 %.

Чтобы избежать накопления ошибок, необходимо применять следующие рекомендации по выбору наилучшей последовательности вычислений [29]:

1. При сложении большого числа слагаемых на микрокалькуляторе следует разбить их на группы чисел, близких по порядку, просуммировать слагаемые каждой группы, а затем складывать полученные суммы последовательно от наименьшей к наибольшей.

2. При умножении нескольких чисел рекомендуется умножать самый большой множитель на самый малый (учитывая результат предыдущего умножения). Это предотвращает попадание в процессе вычислений в область машинного нуля или бесконечности.

3. Избегать вычитания и деления на разность близких по порядку чисел, преобразуя формулу вычислений. Например, проводить преобразования такого рода:

$$(a + b)^2 - a^2 = 2ab + b^2, \frac{1}{\sqrt{a^2 + b^2} - a} = \frac{\sqrt{a^2 + b^2} + a}{b^2},$$

если $a \gg b$.

4. Выбирать лучше тот алгоритм, который требует меньшего количества вычислений.

5. Выбирать по возможности численный метод, устойчивый к погрешностям округлений.

Глава 3. МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

3.1. Решение системы линейных алгебраических уравнений методом Гаусса

Многие задачи инженерной практики и вычислительной математики сводятся к решению систем линейных уравнений. Это задачи из области электротехники, радиоэлектроники, механики, статистики, экономики, а также разностной аппроксимации дифференциальных и интегральных уравнений, локальной линеаризации систем нелинейных алгебраических уравнений и др.

Рассмотрим методы решения системы линейных алгебраических уравнений:

$$Ax = b, \quad (3.1)$$

где $A = [a_{ij}]$ — квадратная матрица порядка n ,

$$b = \begin{pmatrix} a_{1,n+1} \\ \vdots \\ a_{n,n+1} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} — \text{векторы-столбцы.}$$

Если матрица A системы не вырождена, т. е. $\det A \neq 0$, то такая система имеет единственное решение, которое может быть вычислено по формулам Крамера.

Однако применение формул Крамера для решения системы (3.1) в случае больших n ($n \geq 4$) нецелесообразно, поскольку это связано с выполнением слишком большого количества операций и значительным накоплением ошибок округления.

Наиболее известным из точных методов решения системы (3.1) является *метод Гаусса*. Идея метода Гаусса состоит в том, что рассматриваемая система приводится к эквивалентной ей системе с верхней треугольной матрицей (прямой ход). Из преобразованной таким образом системы неизвестные находятся последовательными подстановками, начиная с последнего уравнения преобразованной системы (обратный ход). Алгоритм Гаусса состоит из выполнения однотипных операций, которые легко программируются на современных счетных машинах. Однако точность результата и затрачиваемое на его получение время во многом зависят от организации вычислений. Существует много реализаций схемы Гаусса, дающих те или иные преимущества, в частности, есть схемы с заранее предписанным порядком исключения неизвестных, например, схема единственного деления, или же порядок исключения выбирают по ходу процесса (схема с выбором главного элемента). Описание таких вычислительных схем имеется в литературе [2, 4, 9—11, 14, 19, 27, 32].

Схема единственного деления. Считая $a_{11} \neq 0$ (a_{11} — ведущий элемент), делим первое уравнение системы (3.1) на a_{11}

$$x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n = a_{1,n+1}^{(1)},$$

$$\text{где } a_{1i}^{(1)} = \frac{a_{1i}}{a_{11}}, \quad i = 2, 3, \dots, n+1.$$

Затем исключаем x_1 из остальных уравнений системы, вычитая из i -го уравнения первое, умноженное на соответствующий коэффициент a_{1i} . Приводим преобразованные таким образом уравнения

$$\sum_{j=2}^n a_{ij}^{(1)}x_j = a_{i,n+1}^{(1)}, \quad i = 2, 3, \dots, n, \quad (3.2)$$

$$\text{где } a_{ij}^{(1)} = a_{ij} - a_{1i}a_{1j}^{(1)}, \quad j = 2, 3, \dots, n+1.$$

Неизвестное x_1 оказалось исключенным из всех уравнений, кроме первого. Далее, предположив, что $a_{22}^{(1)} \neq 0$, делим второе уравнение на $a_{22}^{(1)}$ и исключаем неизвестное x_2 из всех уравнений, начиная с третьего, и т. д. до тех пор, пока не переберем все уравнения системы. После таких преобразований получим систему с верхней треугольной матрицей:

Описанный прямой ход метода Гаусса реализуем в том случае, если ведущие элементы $a_{kk}^{(k-1)}$ ($k = 1, 2, \dots, n$) отличны от нуля. Это условие будет выполнено, если отличны от нуля главные миноры матрицы A . Обратный ход метода Гаусса заключается в нахождении неизвестных из преобразованной системы (3.3). Значение x_n определяется из последнего уравнения, x_{n-1} из $n - 1$ -го уравнения и т. д.:

$$x_k = a_{k,n+1}^{(k)} - \sum_{i=k+1}^n a_{ki}^{(k)} x_i \quad (k = n-1, n-2, \dots, 1). \quad (3.4)$$

Количество операций умножения и деления вычисляется по формуле:

$$N = \frac{2n^3 + 9n^2 + n}{6}.$$

При больших n в процессе решения системы желательно производить контроль вычислений. Для этого используются контрольные суммы

$$a_{i,n+2} = \sum_{j=1}^{n+1} a_{ij} \quad (i = 1, 2, \dots, n),$$

которые равны суммам элементов строк матрицы, включая и свободные члены. Если над контрольными суммами в каждой строке проводить те же преобразования, что и над остальными элементами этой строки, то при отсутствии ошибок в вычислениях контрольные суммы должны быть равны суммам элементов соответствующих преобразованных строк расширенной матрицы.

Схема Холецкого. Решение системы линейных уравнений на микрокалькуляторах можно выполнять в автоматическом режиме только в случае небольшого числа уравнений, так

как в памяти хранится $n^2 + n$ элементов расширенной матрицы системы.

При больших n метод Гаусса реализуется в полуавтоматическом режиме, т. е. строки преобразуются последовательно, а промежуточные результаты нужно записывать в таблицу. В таких случаях удобно проводить гауссова исключения по схеме Холецкого [14], позволяющей компактно записывать преобразованные элементы исходной матрицы и упростить программирование процесса вычислений.

В схеме Холецкого матрица системы A представляется в виде произведения двух треугольных матриц — нижней L и верхней U :

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix},$$

$$U = \begin{pmatrix} 1 & u_{12} & \dots & u_{1n} \\ 0 & 1 & \dots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}.$$

Такой алгоритм называют LU -разложением [27].

Элементы l_{ij} , u_{ij} и вектор-столбец \mathbf{y} вычисляются по формулам:

$$l_{ii} = a_{ii} (i = 1, 2, \dots, n), \quad (3.5)$$

$$u_{1j} = \frac{a_{1j}}{a_{11}} (j = 2, 3, \dots, n), \quad y_1 = \frac{a_{1,n+1}}{a_{11}}, \quad (3.6)$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj} (i \geq j > 1), \quad (3.7)$$

$$u_{ij} = \frac{1}{l_{ii}} \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{kj} \right) (1 < i < j), \quad (3.8)$$

$$y_i = \frac{1}{l_{ii}} \left(a_{i,n+1} - \sum_{k=1}^{i-1} l_{ik}y_k \right) (1 < i \leq n). \quad (3.9)$$

Преобразованная расширенная матрица системы имеет вид:

$$\left(\begin{array}{c|ccccc|c} l_{11} & u_{12} & u_{13} & \dots & u_{1n} & y_1 \\ \hline l_{21} & l_{22} & u_{23} & \dots & u_{2n} & y_2 \\ l_{31} & l_{32} & l_{33} & \dots & u_{3n} & y_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} & y_n \end{array} \right). \quad (3.10)$$

Порядок заполнения матрицы (3.10) следующий:

выписываем элементы l_{ii} 1-го столбца (3.5);

вычисляем элементы u_{1i} и y_1 1-й строки (формулы (3.6));

вычисляем элементы l_{i2} 2-го столбца (формулы (3.7));

вычисляем элементы $u_{2,i}$ и y_2 2-й строки (формулы (3.8))

и (3.9)) и т. д. Заполняем всю матрицу в последовательности столбец-строка.

LU-разложение позволяет представить систему (3.1) в виде $LUX = b$, а вектор x определить из последовательного решения систем:

$$Ly = b, \quad Ux = y.$$

Прямой ход заключается в решении системы $L\mathbf{y} = \mathbf{b}$, т. е. в заполнении матрицы (3.10) и получении элементов матрицы U и вектора \mathbf{y} . При обратном ходе вектор \mathbf{x} определяем из системы:

Значения x_i находятся по формулам:

$$x_n = y_n, \\ x_i = y_i - \sum_{k=i+1}^n u_{ik} x_k \quad (i < n). \quad (3.12)$$

Вычисления по формулам (3.6) — (3.9) и (3.12) легко реализуются на микрокалькуляторах.

Схема с выбором главного элемента. Если среди ведущих элементов $a_{kk}^{(k-1)}$ окажется нулевой элемент, то перестановкой строк можно переместить ненулевой элемент в k -м столбце на главную диагональ и продолжить вычисления. Если элемент $a_{kk}^{(k-1)}$ близок к нулю, то деление на него уравнения приводит к большим по модулю преобразованным коэффициентам. В обратном ходе метода Гаусса на эти коэффициенты будут умножаться неточные значения неизвестных, что может привести к росту погрешности и потере точности.

Чтобы избежать существенного влияния вычислительной погрешности, применяют метод Гаусса с частичным выбором главного элемента. В данном методе в качестве ведущего элемента на k -м шаге выбирают наибольший по модулю элемент в k -м столбце и перестановкой строк переводят его на главную диагональ.

В схеме Гаусса с выбором главного элемента ведущий (максимальный по модулю) элемент находят среди всех элементов матрицы, и соответствующее ему значение x_k исключают из всех уравнений, среди коэффициентов которых еще не выбирались главные элементы.

Метод Гаусса с выбором главного элемента прост и численно устойчив для хорошо обусловленных линейных систем (см. п. 3.3).

Схема Гаусса—Жордана, или метод полного исключения [34], заключается в одновременном исключении какого-либо переменного из всех уравнений системы, кроме одного. Его удобно реализовать на микрокалькуляторах, учитывая ограниченность их памяти, так как такая схема вычислений не требует выполнения обратного хода.

На первом шаге этого метода выберем ведущий элемент $a_{11} \neq 0$ (перестановкой уравнений системы можно добиться того, что a_{11} будет наибольшим по модулю коэффициентом при x_1). Разделим первое уравнение системы на a_{11} , во всех остальных уравнениях исключим x_1 , т. е. сведем расширенную матрицу системы к виду:

$$\left(\begin{array}{ccccc|c} 1 & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & a_{2,n+1}^{(1)} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & a_{n,n+1}^{(1)} \end{array} \right),$$

где

$$a_{1j}^{(1)} = a_{1j}/a_{11}, \quad a_{ij}^{(1)} = a_{ij} - a_{1j}^{(1)} \cdot a_{11},$$

$j = 2, 3, \dots, n + 1; i = 2, 3, \dots, n$.

На втором шаге выберем ведущий элемент $a_{22}^{(1)}$ (можно сделать перестановку строк $2, \dots, n$ таким образом, чтобы он был наибольшим по модулю). Разделим второе уравнение на $a_{22}^{(1)}$, исключим x_2 из всех уравнений, кроме второго:

$$\left(\begin{array}{ccccc|c} 1 & 0 & a_{13}^{(2)} & \dots & a_{1n}^{(2)} & a_{1,n+1}^{(2)} \\ 0 & 1 & a_{23}^{(2)} & \dots & a_{2n}^{(2)} & a_{2,n+1}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \dots & a_{3n}^{(2)} & a_{3,n+1}^{(2)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & a_{n3}^{(2)} & & a_{nn}^{(2)} & a_{n,n+1}^{(2)} \end{array} \right),$$

где $a_{2j}^{(2)} = a_{2j}^{(1)}/a_{22}^{(1)}$, $a_{ij}^{(2)} = a_{ij}^{(1)} - a_{2j}^{(2)} \cdot a_{i2}^{(1)}$, $i = 1, 3, \dots, n$; $j = 3, \dots, n + 1$.

После n шагов получим матрицу

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & \dots & 0 & a_{1,n+1}^{(n)} \\ 0 & 1 & 0 & \dots & 0 & a_{2,n+1}^{(n)} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & a_{n,n+1}^{(n)} \end{array} \right)$$

и численные значения неизвестных

$$x_i = a_{i,n+1}^{(n)}, \quad i = 1, 2, \dots, n.$$

Контроль вычислений можно осуществлять так же, как и в схеме единственного деления, используя контрольные суммы.

3.2. Вычисление определителя и обратной матрицы методом Гаусса

В прямом ходе метода Гаусса над элементами матрицы A производятся элементарные преобразования, которые не изменяют определитель матрицы, кроме операции деления на ведущий элемент. Матрица преобразуется к треугольному виду с единичными диагональными элементами, ее определитель равен единице. Если в прямом ходе строки матрицы не переставляются, то знак определителя не изменяется. Таким образом, определитель невырожденной матрицы системы равен произведению ведущих элементов в прямом ходе исключения Гаусса

$$\Delta = \det A = a_{11}a_{22}^{(1)} \dots a_{nn}^{(n-1)}. \quad (3.13)$$

Для его вычисления осуществляем прямой ход метода Гаусса, как и при решении системы, только без преобразований вектора b . При решении линейной системы определитель можно вычислить попутно.

Если применяется метод исключения с выбором главного элемента, то в (3.13) необходимо добавить множитель $(-1)^k$, где k — количество перестановок строк и столбцов.

Все вычислительные схемы метода Гаусса позволяют осуществлять одновременное решение систем линейных уравнений с различными правыми частями: $A\mathbf{x} = \mathbf{b}^{(k)}$, $k = 1, 2, \dots, m$. При этом количество вычислений увеличивается только на преобразование новых столбцов правых частей, что дает значительную экономию времени счета.

В частности, если вместо столбцов $\mathbf{b}^{(k)}$ выбирать столбцы единичной матрицы порядка n : $\mathbf{e}^{(k)} = (0, 0, \dots, 1, \dots, 0)$ (единица на k -м месте, остальные элементы — нули), $k = 1, 2,$

..., n , то решение $x^{(k)}$ системы $Ax = e^{(k)}$ будет k -м столбцом обратной матрицы A^{-1} .

Таким образом, для вычисления обратной матрицы требуется решить одновременно n систем уравнений с n неизвестными. Проводим последовательные исключения неизвестных в расширенной матрице

$$\left(\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & 0 & 0 & \cdots & 1 \end{array} \right) \underset{\substack{\sim \\ (\text{после } n \\ \text{шагов})}}{\sim} \left(\begin{array}{cccc|cccc} 1 & 0 & \cdots & 0 & x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(n)} \\ 0 & 1 & \cdots & 0 & x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(n)} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & x_n^{(1)} & x_n^{(2)} & \cdots & x_n^{(n)} \end{array} \right).$$

Справа получим элементы обратной матрицы

$$A^{-1} = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(n)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(n)} \\ \vdots & \ddots & \ddots & \vdots \\ x_n^{(1)} & x_n^{(2)} & \cdots & x_n^{(n)} \end{pmatrix}.$$

3.3. Погрешность решения системы линейных уравнений

Введем понятия нормы вектора и матрицы, а также понятие обусловленности матрицы, с которыми связаны вопросы оценки погрешности решения системы линейных уравнений [10, 20, 21].

Рассмотрим линейное пространство R^n n -мерных векторов $\mathbf{x} = (x_1, x_2, \dots, x_n)$.

Нормой вектора x называют вещественное число $\|\mathbf{x}\|$, удовлетворяющее условиям:

- 1) $\|\mathbf{x}\| \geq 0$, причем $\|\mathbf{x}\| = 0$ тогда и только тогда, если $\mathbf{x} = 0$;
- 2) $\|c\mathbf{x}\| = |c| \cdot \|\mathbf{x}\|$, где c — число;
- 3) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.

В частности, функция

$$\|\mathbf{x}\|_p = \left(\sum_{k=1}^n |x_k|^p \right)^{\frac{1}{p}}$$

при любом $p \geq 1$ удовлетворяет аксиомам нормы. Норма

такого типа называется *нормой Гёльдера с показателем* p . Среди гёльдеровых норм наиболее распространенными являются следующие:

$$\begin{aligned}\|\boldsymbol{x}\|_1 &= \sum_{k=1}^n |x_k|, \quad \|\boldsymbol{x}\|_2 = \left(\sum_{k=1}^n |x_k|^2 \right)^{\frac{1}{2}}, \\ \|\boldsymbol{x}\|_\infty &= \max_{1 \leq k \leq n} |x_k|. \end{aligned} \quad (3.14)$$

Норма $\|\boldsymbol{x}\|_2$ называется также евклидовой и обозначается $\|\boldsymbol{x}\|_E$.

Все нормы векторов равнозначны в том смысле, что если последовательность векторов $\{\boldsymbol{x}^k = (x_1^k, x_2^k, \dots, x_n^k)\}$ сходится по некоторой норме к вектору $\boldsymbol{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$, т. е.

$$\lim_{k \rightarrow \infty} \|\boldsymbol{x}^k - \boldsymbol{x}^0\| = 0,$$

то она сходится к вектору \boldsymbol{x}^0 и по любой другой норме.. В конечномерном нормированном пространстве из сходимости по норме вытекает покоординатная сходимость и наоборот. При этом предполагается, что последовательность векторов $\{\boldsymbol{x}^k\}$ сходится покоординатно к вектору \boldsymbol{x}^0 , если для всех $i = 1, 2, \dots, n$ выполнены соотношения

$$\lim_{k \rightarrow \infty} x_i^k = x_i^0.$$

Множество матриц одинаковой размерности является линейным пространством. Введем в этом пространстве норму матрицы таким образом, чтобы ее можно было рассматривать и как норму оператора.

Нормой матрицы A называется вещественное число $\|A\|$, удовлетворяющее следующим аксиомам:

1. $\|A\| \geq 0$, $\|A\| = 0$ тогда и только тогда, если $A = \theta$ (θ — нулевая матрица);
- 2) $\|\alpha A\| = |\alpha| \cdot \|A\|$, где α — число;
- 3) $\|A + B\| \leq \|A\| + \|B\|$.

Норма матрицы $\|A\|$ называется *согласованной* с нормой вектора $\|\boldsymbol{x}\|$, если для любых A и \boldsymbol{x}

$$\|A\boldsymbol{x}\| \leq \|A\| \cdot \|\boldsymbol{x}\|.$$

Отметим, что с одной и той же нормой вектора могут быть согласованы различные нормы матриц.

Пусть $\|A\|$ — норма матрицы, согласованная с заданной нормой вектора $\|\boldsymbol{x}\|$. Норма $\|A\|$ называется *подчиненной*

норме вектора $\|\mathbf{x}\|$, если найдется такой вектор $\mathbf{x} \neq 0$, что

$$\|A\mathbf{x}\| = \|A\| \cdot \|\mathbf{x}\|,$$

т. е.

$$\|A\| = \sup_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|. \quad (3.15)$$

Таким образом, среди всех норм, согласованных с заданной векторной нормой, подчиненная норма является минимальной. Отметим, что для любой нормы вектора имеется хотя бы одна подчиненная норма матрицы.

Любая норма матрицы удовлетворяет неравенствам:

$\|E\| \geq 1$, где E — единичная матрица;

$\|A\| \cdot \|A^{-1}\| \geq 1$, A — невырожденная матрица.

Для подчиненной нормы выполняются соотношения

$$\|E\| = 1; \|(E \pm A)^{-1}\| \leq \frac{1}{1 - \|A\|}; \|AB\| \leq \|A\| \cdot \|B\|. \quad (3.16)$$

При этом предполагается, что соответствующие операции имеют смысл.

Наиболее употребительны следующие нормы матриц:

$$\|A\|_1 = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|. \quad (3.17)$$

Эта норма матрицы A ($n \times m$) подчинена норме вектора $\|\mathbf{x}\|_1$. Спектральная норма квадратной матрицы

$$\|A\|_2 = \max_i \sigma_i, \quad (3.18)$$

где $\sigma_i = \sqrt{\mu_i}$ — сингулярные числа матрицы A , а μ_i — собственные числа матрицы $A^T A$ (A^T — транспонированная матрица). Если матрица A — симметрическая, т. е. $A^T = A$, то $\|A\|_2 = \max_i |\lambda_i|$, где λ_i — собственные числа матрицы A .

Спектральная норма матрицы подчинена евклидовой норме вектора $\|\mathbf{x}\|_2$.

Норма

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^m |a_{ij}| \quad (3.19)$$

подчинена норме вектора $\|\mathbf{x}\|_\infty$.

Евклидова норма матрицы

$$\|A\|_E = \left(\sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2 \right)^{\frac{1}{2}} \quad (3.20)$$

согласована с нормой вектора $\|\mathbf{x}\|_2$. В общем случае евклидова норма матрицы не является подчиненной.

Известно [21], что для симметрической матрицы A среди всех возможных норм $\|A\|$, согласованных с некоторой нормой вектора, спектральная норма $\|A\|_2$ будет минимальной.

Пусть ненулевой вектор x является собственным вектором матрицы A , которому соответствует собственное число λ , т. е. $Ax = \lambda x$. Тогда для любой согласованной нормы $\|A\|$ получим $|\lambda| \cdot \|x\| = \|\lambda x\| = \|Ax\| < \|A\| \cdot \|x\|$, откуда

$$|\lambda| < \|A\|. \quad (3.21)$$

Таким образом, модули собственных чисел матрицы A не превышают любой ее согласованной нормы.

В практических задачах элементы матрицы A и вектора b в системе (3.1) суть приближенные числа. При решении системы произвольным точным методом неизбежно появляются ошибки округления. Установим зависимость между погрешностью решения и свойствами матрицы A .

Будем интерпретировать погрешности вычислений как погрешности, вызванные возмущением правой части. Пусть $\hat{x} = x + \Delta x$ — решение возмущенной системы $A(x + \Delta x) = b + \Delta b$, где Δb — возмущение вектора b , Δx — соответствующая погрешность точного решения x . Тогда

$$Ax + A\Delta x = b + \Delta b; A\Delta x = \Delta b; \Delta x = A^{-1}\Delta b,$$

$$\|\Delta x\| \leq \|A^{-1}\| \cdot \|\Delta b\|.$$

Учитывая, что $b = Ax$ и $\|b\| \leq \|A\| \cdot \|x\|$, можно записать $\|\Delta x\| \cdot \|\Delta b\| \leq \|A\| \cdot \|A^{-1}\| \cdot \|x\| \cdot \|\Delta b\|$. Отсюда следует

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\Delta b\|}{\|b\|}.$$

Таким образом, относительная погрешность решения $\delta_x = \frac{\|\Delta x\|}{\|x\|}$ оценивается через относительную погрешность правой части $\delta_b = \frac{\|\Delta b\|}{\|b\|}$ посредством неравенства

$$\delta_x \leq \|A\| \cdot \|A^{-1}\| \cdot \delta_b. \quad (3.22)$$

Величина

$$\operatorname{cond}(A) = \|A\| \cdot \|A^{-1}\| \quad (3.23)$$

называется мерой или числом обусловленности матрицы A . Она является максимально возможным коэффициентом усиления ($\operatorname{cond}(A) \geq 1$) относительной погрешности решения при возмущении правой части. Аналогичные результаты имеют место при изменении коэффициентов матрицы.

В частности, если матрица A — симметрическая и выбрана спектральная норма матрицы $\|A\|_2$, то

$$\|A\|_2 = \max_i |\lambda_i(A)|, \|A^{-1}\|_2 = \max_i \frac{1}{|\lambda_i(A)|} = \frac{1}{\min_i |\lambda_i(A)|},$$

и число обусловленности равно

$$\operatorname{cond}(A) = \frac{\max_i |\lambda_i(A)|}{\min_i |\lambda_i(A)|}. \quad (3.24)$$

Как следует из неравенства (3.22), погрешность решения системы (3.1) может оказаться значительной, если матрица системы характеризуется большим разбросом собственных чисел.

Матрицу называют *плохо обусловленной*, если соответствующая ей обратная матрица будет неустойчивой, т. е. малым изменениям в элементах матрицы соответствуют значительные изменения в элементах обратной матрицы [32].

Иногда в качестве меры отклонения точного решения \mathbf{x} от вычисленного $\tilde{\mathbf{x}} = \mathbf{x} + \Delta \mathbf{x}$ принимают невязку:

$$\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}.$$

Однако в общем случае из малости невязки не следует малость ошибки $\Delta \mathbf{x}$ [33]. Но если система хорошо обусловлена, то малая невязка свидетельствует об удовлетворительной точности решения.

Погрешность решения будет незначительной для систем, у которых диагональные элементы матрицы преобладают над остальными.

Решение системы линейных уравнений методом Гаусса с выбором главного элемента обеспечивает малые невязки и в случае хорошей обусловленности системы метод будет устойчивым.

В настоящее время для больших ЭВМ составлены программы решения систем линейных уравнений, в которых учитывается оценка числа обусловленности матрицы [33].

3.4. Лабораторная работа 1. Решение системы линейных алгебраических уравнений методом Гаусса—Жордана

Программа 1 решения системы трех линейных алгебраических уравнений по схеме Гаусса—Жордана в автоматическом режиме вычислений:

Адрес	Команда	Код	Содержание операций
00	ИП1	61	
01	ИП2	62	
02	ПП	53	
03	15	15	
04	П8	48	
05	ИП3	63	
06	П9	49	
07	ПП	53	
08	31	31	
09	ПП	53	
10	15	15	
11	ПП	53	
12	23	23	
13	П2	42	Засылка x_2 в регистр 2
14	С/П	50	Останов; индикация x_2
15	ПП	53	
16	23	23	
17	ПВ	4L	
18	ИП3	63	
19	ПС	4E	
20	ПП	53	
21	31	31	
22	В/О	52	
23	П2	42	
24	F,	25	
25	П1	41	
26	ИП3	63	
27	ПД	4F	
28	1	01	
29	3	03	
30	П0	40	
31	КИП0	10	
32	КИП0	10	
33	КИП0	10	
34	ИП1	61	
35	÷	13	
36	П3	43	
37	ИП2	62	
38	×	12	
39	—	11	
40	КП↑	LE	
41	ХУ	14	
42	ИП3	63	
43	ИПД	6F	
44	×	12	
45	—	11	
46	В/О	52	

Инструкция. Элементы расширенной матрицы системы засыпаем в регистры по схеме:

$$\begin{pmatrix} 1 & 4 & 7 & A \\ 2 & 5 & 8 & B \\ 3 & 6 & 9 & C \end{pmatrix}.$$

После вычислений (В/О С/П) значения x_1 , x_2 , x_3 хранятся соответственно в регистрах: А и Y, 2 и X, 3. Время счета — 45 с.

Контрольный пример. Решить систему

$$\begin{cases} 10x_1 - 3,99x_2 + 7,99x_3 = 15,40, \\ 2,50x_1 - 10x_2 - 2,50x_3 = -11, \\ 7,99x_1 + 3,99x_2 + 10x_3 = 24,17 \end{cases}$$

методом Жордана—Гаусса.

В режиме программирования (F ПРГ) вводим программу. В режиме автоматической работы (F АВТ) вводим элементы расширенной матрицы системы:

10 П1 П9 /—/ П5 2,5 П2 /—/ П8 7,99 П3 П7
3,99 П6 /—/ П4 15,4 ПА 11 /—/ ПВ 24,17 ПС

Нажимаем клавиши В/О, С/П. После останова записываем результат: $x_2 = 1,2333263$, (ИПА) $x_1 = 1,3664304$, (ИПЗ) $x_3 = 0,83312482$. Вычислим невязки $r_1 = r_3 = 10^{-6}$, $r_2 = -10^{-6}$. Округлив результаты до двух десятичных знаков, запишем ответ: $x_1 = 1,37$; $x_2 = 1,23$; $x_3 = 0,83$.

Программа 2 одного шага метода Гаусса—Жордана с выбором главного элемента для решения системы n уравнений ($n \leq 9$) в полуавтоматическом режиме:

ПВ П0 С/П ПД С/П КП↑ FL0 04 С/П ИПД
 ÷ ПС ИПВ П0 С/П КИП↑ ИПС × — FL0
 14 БП 08

Распределение регистров памяти:

Номер регистра	0	1	...	A	B	C	Д
Переменная	счетчик цикла	$a_{j,n+1}$		a_{j1}	n	a_{ik}/a_{jk}	a_{jk}

Инструкция.

1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) ввести исходные данные: n , элементы ведущей (j -й) строки и ведущий (k -й) элемент по схеме:

$$(n) \text{ В/О С/П } (a_{jk}) \text{ С/П } (a_{j1}) \text{ С/П } \dots (a_{j, k-1}) \text{ С/П} \\ (a_{j, k+1}) \text{ С/П } \dots (a_{j, n+1}) \text{ С/П},$$

т. е. ведущий элемент ввести первым.

3. Ввести k -й элемент ведомой строки: (a_{ik}) С/П.

4. Вводить остальные элементы ведомой строки:

$$(a_{i1}) \text{ С/П } \dots (a_{i, k-1}) \text{ С/П } (a_{i, k+1}) \text{ С/П } \dots (a_{i, n+1}) \text{ С/П}.$$

После каждого из ОСТАНОВов записать с индикатора значение преобразованного элемента.

Для преобразования следующей ведомой строки перейти к п. 3.

Контрольный пример.

Решить методом Гаусса—Жордана с выбором главного элемента систему

$$\begin{cases} 3,1x_1 + 2,7x_2 - 3,3x_3 + 0,6x_4 = 3,2, \\ 1,2x_1 + 3,8x_2 + 4,2x_3 + 2,4x_4 = 14,7, \\ 2,3x_1 + 1,1x_2 + 3,7x_3 + 1,1x_4 = 10,2, \\ 0,4x_1 + 2,5x_2 - 2,0x_3 + 3,4x_4 = 5,6. \end{cases}$$

Вводим программу, нажав клавиши F ПРГ и клавиши программы.

I шаг. Ведущим (наибольшим по абсолютной величине) является элемент $a_{23} = 4,2$. Выполняем п. 2 инструкции по работе с программой: F АВТ 4 В/О С/П 4,2 С/П 1,2 С/П 3,8 С/П 2,4 С/П 14,7 С/П. Для преобразования первой строки вводим значение $(a_{13} = -3,3)$ С/П и осуществляем вычисления по схеме:

3,1 С/П (списываем $a_{11}^{(1)} = 4,0429$), 2,7 С/П (списываем $a_{12}^{(1)} = 5,6857$), 0,6 С/П (списываем $a_{14}^{(1)} = 2,4857$), 3,2 С/П (списываем $a_{15}^{(1)} = 14,7500$).

Аналогично преобразуем третью и четвертые строки.

Результаты удобно заносить в табл. 3.1.

II шаг. Ведущим элементом является $a_{12}^{(1)} = 5,6857$. Отметим, что коэффициенты при x_3 преобразовываться не будут. Введем $n = 3$, ведущий элемент $a_{12}^{(1)}$ и остальные элементы первой строки: 5,6857 С/П 4,0429 С/П 2,4857 С/П 14,7500 С/П. Преобразуем элементы второй строки: 3,8 С/П 1,2 С/П (списываем $a_{21}^{(2)} = -1,5020$) 2,4 С/П (списываем $a_{24}^{(2)} = 0,7387$) 14,7 С/П (списываем $a_{25}^{(2)} = 4,8419$). Аналогично преобразуем третью и четвертую строки. Результаты заносим в табл. 3.1.

III шаг. Ведущим элементом является $a_{31}^{(2)} = 2,8411$.

IV шаг. Ведущий элемент — $a_{44}^{(3)} = 2,6355$.

Таблица 3.1

Номер шага	x_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}
0	3,1 1,2 2,3 0,4	2,7 3,8 1,1 2,5	-3,3 4,2 3,7 -2,0	0,6 2,4 1,1 3,4	3,2 14,7 10,2 5,6
I	4,0429 1,2 1,2429 0,9714	5,6857 3,8 -2,2476 4,3095	0 4,2 0 0	2,4857 2,4 -1,0143 4,5429	14,7500 14,7 -2,7500 12,6000
II	4,0429 -1,5020 2,8411 -2,0930	5,6857 0 0 0	0 4,2 0 0	2,4857 0,7387 -0,0317 2,6589	14,7500 4,8419 3,0808 1,4202
III	0 0 2,8411 0	5,6857 0 0 0	0 4,2 0 0	2,5308 0,7219 -0,0317 2,6355	10,3660 6,4706 3,0808 3,6898
IV	0 0 2,8411 0	5,6857 0 0 0	0 4,2 0 0	0 0 0 2,6355	6,8228 5,4599 3,1252 3,6898
	1 1	1	1	1	$x_2 = 1,2000$ $x_3 = 1,3000$ $x_1 = 1,1000$ $x_4 = 1,4000$

Искомые решения x_i можно определить на IV шаге в процессе вычисления коэффициентов $a_{k,n+1}$, разделив их на соответствующие ведущие элементы. Промежуточные данные округляем до четвертого знака после запятой, результат — до первого знака.

Ответ: $x_1 = 1,1$; $x_2 = 1,2$; $x_3 = 1,3$; $x_4 = 1,4$.

Время счета и ввода 2 мин.

Отметим, что по результатам выполнения IV шага легко вычислить определитель исходной матрицы. Если осуществить две перестановки (первого и второго столбцов, второй и третьей строк), то преобразованная матрица будет диагональной, поэтому $\det A = (-1)^2 \cdot 5,6857 \cdot 4,2 \cdot 2,8411 \times 2,6355 \approx 178,8$.

Задания к лабораторной работе 1. Решить приведенные ниже системы методом Гаусса—Жордана:

- 1) $5,7x_1 + 3,3x_2 + 1,3x_3 = 2,1,$
 $3,5x_1 + 4,7x_2 + 2,1x_3 = 1,7,$
 $4,1x_1 + 5,8x_2 - 11,7x_3 = 0,8;$
- 2) $4,7x_1 + 2,8x_2 + 1,9x_3 = 0,7,$
 $2,1x_1 + 3,4x_2 + 1,8x_3 = 1,1,$
 $4,2x_1 - 1,7x_2 + 9,3x_3 = 2,8;$
- 3) $3,1x_1 + 2,8x_2 + 1,9x_3 = 0,2,$
 $1,9x_1 + 3,1x_2 + 2,1x_3 = 2,1,$
 $7,5x_1 + 3,8x_2 + 9,8x_3 = 5,6;$
- 4) $9,1x_1 + 5,6x_2 + 7,8x_3 = 9,8,$
 $3,8x_1 + 5,1x_2 + 2,8x_3 = 6,7,$
 $4,1x_1 + 5,7x_2 + 7,2x_3 = 5,8;$
- 5) $3,3x_1 + 2,1x_2 + 2,8x_2 = 0,8,$
 $4,1x_1 + 8,7x_2 + 4,8x_3 = 5,7,$
 $2,7x_1 + 1,8x_2 + 7,1x_3 = 3,2;$
- 6) $7,6x_1 + 5,8x_2 + 4,7x_3 = 10,1,$
 $3,8x_1 + 4,1x_2 + 2,7x_3 = 9,7,$
 $2,9x_1 + 2,1x_2 + 3,8x_3 = 7,8;$
- 7) $3,8x_1 - 2,5x_2 + 3,7x_3 = 6,5,$
 $0,5x_1 + 0,84x_2 + 1,7x_3 = 0,$
 $1,6x_1 + 2,3x_2 - 4,5x_3 = 4,3;$
- 8) $5,4x_1 - 2,3x_2 + 3,4x_3 = -3,5,$
 $4,2x_1 + 7,7x_2 - 2,3x_3 = 2,7,$
 $3,4x_1 + 2,4x_2 + 7,8x_3 = 1,9;$
- 9) $8,6x_1 + 1,8x_2 - 4,7x_3 = 3,8,$
 $2,7x_1 - 3,6x_2 + 1,9x_3 = 0,4,$
 $1,5x_1 + 4,5x_2 + 9,3x_3 = -1,6;$
- 10) $5,6x_1 + 2,7x_2 - 1,7x_3 = 1,9,$
 $3,4x_1 - 9,6x_2 - 6,7x_3 = -2,4,$
 $0,8x_1 + 1,3x_2 + 3,7x_3 = 1,2;$
- 11) $2,7x_1 + 0,9x_2 - 1,5x_3 = 3,5,$
 $4,5x_1 - 8,8x_2 + 6,7x_3 = 2,6,$
 $5,1x_1 + 3,7x_2 - 7,4x_3 = -0,14;$
- 12) $9,5x_1 - 3,5x_2 + 7,4x_3 = 2,5,$
 $3,1x_1 - 8,6x_2 - 2,3x_3 = -1,5,$
 $0,8x_1 + 7,4x_2 - 9,5x_3 = 6,4;$
- 13) $8,8x_1 + 6,7x_2 - 1,2x_3 = 5,2,$
 $6,4x_1 + 11,3x_2 - 2,7x_3 = 3,8,$
 $2,4x_1 - 4,5x_2 + 8,5x_3 = -0,6;$
- 14) $9,4x_1 - 6,2x_2 - 0,5x_3 = 0,52,$
 $3,4x_1 + 8,3x_2 + 0,8x_3 = -0,8,$
 $2,4x_1 - 1,1x_2 + 3,8x_3 = 1,8;$
- 15) $7,8x_1 + 5,3x_2 + 4,8x_3 = 1,8,$
 $3,3x_1 + 7,1x_2 + 1,8x_3 = 2,3,$
 $4,5x_1 + 3,3x_2 + 2,8x_3 = 3,4;$
- 16) $3,8x_1 + 4,1x_2 - 2,3x_3 = 4,8,$
 $-2,1x_1 + 3,9x_2 - 5,8x_3 = 3,3,$
 $1,8x_1 + 1,1x_2 - 2,1x_3 = 5,8;$
- 17) $1,7x_1 - 2,2x_2 + 3,0x_3 = 1,8,$
 $2,1x_1 + 1,9x_2 - 2,3x_3 = 2,8,$
 $4,2x_1 + 3,9x_2 - 3,1x_3 = 5,1;$
- 18) $5,8x_1 + 3,3x_2 - 3,2x_3 = 4,5,$
 $2,5x_1 - 7,8x_2 + 3,3x_3 = 7,1,$
 $6,5x_1 - 7,1x_2 + 9,8x_3 = 6,1;$
- 19) $7,1x_1 + 6,8x_2 + 6,1x_3 = 7,0,$
 $5,0x_1 + 9,8x_2 + 5,3x_3 = 6,1,$
 $8,2x_1 + 7,8x_2 + 8,1x_3 = 5,8;$
- 20) $5,0x_1 + 3,1x_2 + 4,0x_3 = 5,0;$
 $4,0x_1 + 4,5x_2 - 4,0x_3 = 4,9,$
 $-2,0x_1 - 3,7x_2 + 6,0x_3 = 2,7;$
- 21) $9,0x_1 + 5,2x_2 - 5,0x_3 = 7,0,$
 $3,8x_1 - 6,0x_2 + 4,0x_3 = 5,0,$
 $7,8x_1 + 5,0x_2 - 9,0x_3 = 5,0;$
- 22) $8,7x_1 - 2,0x_2 + 4,5x_3 = 2,0,$
 $2,0x_1 + 4,0x_2 - 1,8x_3 = 3,0,$
 $1,6x_1 + 5,0x_2 + 6,3x_3 = -2,0;$
- 23) $6,3x_1 + 5,0x_2 - 0,6x_3 = 1,0,$
 $3,4x_1 - 7,0x_2 + 3,4x_3 = 2,0,$
 $0,8x_1 + 1,0x_2 + 3,5x_3 = -2,0;$
- 24) $4,5x_1 + 2,0x_2 - 3,7x_3 = 4,0,$
 $2,1x_1 + 8,0x_2 + 5,8x_3 = -3,0,$
 $x_1 + 1,1x_2 - 2,5x_3 = 5,0;$
- 25) $1,2x_1 + 0,3x_2 - 0,5x_3 = 1,7,$
 $1,3x_1 + 5,7x_2 - 1,2x_3 = 7,8,$
 $0,8x_1 + 4,1x_2 + 9,1x_3 = 5,4;$
- 26) $6,2x_1 + 1,3x_2 - 0,7x_3 = 2,3,$
 $0,4x_1 + 5,7x_2 + 1,3x_3 = 1,9,$
 $2,1x_1 + 0,7x_2 + 3,4x_3 = 1,0;$
- 27) $5,3x_1 - 0,7x_2 + 1,1x_3 = 5,0,$
 $1,2x_1 + 6,1x_2 - 1,3x_3 = 6,0,$
 $2,1x_1 - 1,4x_2 + 9,7x_3 = 10,0;$
- 28) $4,8x_1 - 1,2x_2 + 2,4x_3 = 5,1,$
 $2,7x_1 + 9,8x_2 + 3,1x_3 = 10,2,$
 $0,8x_1 - 0,3x_2 + 3,7x_3 = 4,7;$

$$29) \begin{aligned} 6,7x_1 + 1,3x_2 - 2,7x_3 &= 8,1, \\ 1,3x_1 + 5,7x_2 + 1,3x_3 &= 5,1, \\ 3,1x_1 + 2,3x_2 - 5,4x_3 &= 6,3; \end{aligned}$$

$$30) \begin{aligned} 5,8x_1 + 0,9x_2 + 1,1x_3 &= 6,4, \\ 0,2x_1 + 6,4x_2 - 0,5x_3 &= 3,1, \\ 0,8x_1 - 0,4x_2 + 2,7x_3 &= 3,2. \end{aligned}$$

3.5. Метод прогонки

Большинство технических задач сводится к решению систем линейных алгебраических уравнений, в которых матрицы содержат много нулевых элементов, а ненулевые элементы расположены по специальной структуре, например, ленточные, квазитреугольные матрицы.

Задачи построения интерполяционных сплайнов, разностные методы решения краевых задач для дифференциальных уравнений сводятся к решению системы алгебраических уравнений с трехдиагональной матрицей A . В матрице A все элементы, не лежащие на главной диагонали и двух соседних параллельных диагоналях, равны нулю. Такие системы можно записать

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i, \quad 1 \leq i \leq n; \quad a_1 = 0; \quad c_n = 0. \quad (3.25)$$

Выбор наибольшего элемента при исключении неизвестных методом Гаусса в таких системах делать нельзя, поскольку перестановка строк разрушает структуру матрицы. Наиболее часто к решению систем с трехдиагональной матрицей применяют метод прогонки, который является частным случаем метода Гаусса.

Прямой ход метода прогонки заключается в исключении элементов a_i в системе (3.25). Так как $a_1 = 0$, то первое уравнение системы примет вид

$$b_1 x_1 + c_1 x_2 = d_1.$$

Выразим x_1 через x_2 : $x_1 = -\frac{c_1}{b_1} x_2 + \frac{d_1}{b_1}$ и подставим во второе уравнение системы. Получим уравнение, связывающее x_2 и x_3 , и т. д. Пусть уже получено соотношение

$$x_i = k_i x_{i+1} + \varphi_i \quad (i = 1, 2, \dots, n). \quad (3.26)$$

Понизим в (3.26) индекс на единицу и подставим значение x_{i-1} в i -е уравнение системы (3.25):

$$a_i k_{i-1} x_i + a_i \varphi_{i-1} + b_i x_i + c_i x_{i+1} = d_i.$$

Отсюда

$$x_i = \frac{-c_i}{b_i + a_i k_{i-1}} x_{i+1} + \frac{-a_i \varphi_{i-1} + d_i}{b_i + a_i k_{i-1}}.$$

Слагивая это выражение с (3.26), получим рекуррентные формулы для вычисления k_i и φ_i в прямом ходе:

$$k_i = \frac{-c_i}{b_i + a_i k_{i-1}}; \quad \varphi_i = \frac{-a_i \varphi_{i-1} + d_i}{b_i + a_i k_{i-1}}. \quad (3.27)$$

Учитывая, что $a_1 = c_n = 0$, полагаем $k_0 = \varphi_0 = x_{n+1} = 0$. Обратный ход осуществляется по (3.26).

Почти во всех задачах, приводящих к решению системы (3.25) с трехдиагональной матрицей, обеспечивается условие преобладания диагональных элементов

$$|b_i| \geq |a_i| + |c_i|.$$

Это обеспечивает существование единственного решения и достаточно хорошую устойчивость метода прогонки относительно ошибок округления.

Программа 3 решения системы n линейных уравнений с трехдиагональной симметрической матрицей методом прогонки ($n \leq 5$) в автоматическом режиме:

↑	З	Х	2	—	П0	0	П6	БП	16
КП0	ХY	С/П	Х	КП0	ВХ	↑	F,	Х	С/П
—	/—/	1/x	С/П	F,	F,	×	—	Х	FL0
10	С/П	КИП6	Х	КИП6	—	/—/	КИП6	F,	БП
31									

Распределение регистров памяти:

Номер регистра	0,6	1, 4, 7, А	2, 5, 8, В
Переменная	счетчики циклов	k_{n-1}, \dots, k_1	$\varphi_{n-1}, \dots, \varphi_1$

Инструкция.

После ввода программы в режиме программирования переходим в режим автоматической работы и осуществляем ввод коэффициентов и вычисления:

(n) В/О С/П (b_1) С/П (d_1) С/П (a_2) С/П (b_2) С/П
 (d_2) С/П ... (a_n) С/П (b_n) С/П (d_n) С/П

(списываем значение x_n) С/П (списываем значение x_{n-1}) ... С/П (списываем значение x_1).

Контрольный пример.

Решить методом прогонки систему

$$\left\{ \begin{array}{l} 4x_1 + x_2 = 5,6, \\ x_1 + 4x_2 + x_3 = 7,2, \\ x_2 + 4x_3 + x_4 = 7,8, \\ x_3 + 4x_4 + x_5 = 8,4, \\ x_4 + 4x_5 = 7,4. \end{array} \right.$$

После ввода программы нажимаем клавиши

F	ABT	5	B/O	C/P	4	C/P	5,6	C/P	1	C/P
4			C/P	7,2	C/P	1	C/P	4	C/P	
1			C/P	4	C/P	8,4	C/P	1	C/P	7,4

(списываем $x_5 = 1.4999999$) C/P (списываем $x_4 = 1.4000001$) C/P (спи-
сызаем $x_3 = 1.3$) C/P (списываем $x_2 = 1.1999999$) C/P (спи-
сызаем $x_1 = 1.1$).

Время ввода коэффициентов и счета ~ 3 мин.

Ответ: $x_1 = 1.1$; $x_2 = 1.2$; $x_3 = 1.3$; $x_4 = 1.4$; $x_5 = 1.5$.

Программа 4 решения системы линейных уравнений произвольного порядка n с трехдиагональной матрицей методом прогонки в полуавтоматическом режиме:

C _x	PA	C/P	ИПА	ИПС	×	—	ПВ	C/P	ИПВ
÷	ПС	C/P	ПП	27	БП	01	ПП	27	XU
ИПД	×	—	ПД	C/P	БП	19	ИПД	ИПА	×
—	ИПВ	÷	ПД	C/P	В/O				

Распределение регистров памяти:

Номер регистра	A	B	C	D
Переменная	a_i	b_i	c_i	d_i

Инструкция.

1. В режиме программирования (F ПРГ) ввести про-
грамму.

2. В автоматическом режиме (F ABT) нажать клавиши
B/O C/P и вводить коэффициенты системы построчно, нажимая
после каждого ввода клавишу C/P. После ввода коэффициен-
тов c_i и d_i записать их преобразованные значения c'_i и d'_i .
Коэффициенты заносятся в такой последовательности:

B/O C/P (b_1) C/P (c_1) C/P (записать c'_1) (d₁)

C/P (записать d'_1) (a₂) C/P (b_2) C/P (c_2) C/P (записать c'_2)
(d₂) C/P (записать d'_2) ... (a_n)
C/P (b_n) C/P (d_n) БП 17 C/P.

3. После ОСТАНОВа выполнить обратный ход:
 записать x_n , ввести $(c'_{n-1}) \uparrow (d'_{n-1})$ С/П,
 записать x_{n-1} , ввести $(c'_{n-2}) \uparrow (d'_{n-2})$ С/П,

 записать x_2 , ввести $(c'_1) \uparrow (d'_1)$ С/П,
 записать x_1 .

3.6. Итерационные методы решения линейных систем

Итерационные методы решения системы (3.1) основаны на выборе некоторого начального приближения $x^{(0)}$ с последующим его уточнением по рекуррентным формулам до получения приближения $\mathbf{x}^{(k)}$ с заданной точностью. Процесс итераций легко реализуется на ЭВМ. Он самоисправляемый, т. е. отдельные ошибки в вычислениях некоторого приближения не отражаются на окончательном результате решения.

Сходимость итерационного процесса можно доказать, исходя из общего принципа сжатых отображений [20, 21].

Принцип сжатых отображений. Множество X называется *метрическим пространством*, если каждой паре элементов $\mathbf{x}, \mathbf{y} \in X$ поставлено в соответствие неотрицательное вещественное число $\rho(\mathbf{x}, \mathbf{y})$ (расстояние), удовлетворяющее следующим аксиомам:

- 1) $\rho(\mathbf{x}, \mathbf{y}) \geq 0$, $\rho(\mathbf{x}, \mathbf{y}) = 0$ тогда и только тогда, если $\mathbf{x} = \mathbf{y}$;
- 2) $\rho(\mathbf{x}, \mathbf{y}) = \rho(\mathbf{y}, \mathbf{x})$ (аксиома симметрии);
- 3) $\rho(\mathbf{x}, \mathbf{y}) \leq \rho(\mathbf{x}, \mathbf{z}) + \rho(\mathbf{z}, \mathbf{y})$ для любых элементов $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$ (неравенство треугольника).

Элементы метрического пространства называются *точками*.

Элемент $\mathbf{x}^{(0)}$ метрического пространства X называется пределом последовательности $\{\mathbf{x}^{(k)}\}$ точек $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(k)}, \dots$, принадлежащих X , если последовательность расстояний $\rho(\mathbf{x}^{(0)}, \mathbf{x}^{(k)})$ сходится к нулю при $k \rightarrow \infty$, т. е.

$$\lim_{k \rightarrow \infty} \rho(\mathbf{x}^{(0)}, \mathbf{x}^{(k)}) = 0.$$

Последовательность $\{\mathbf{x}^{(k)}\}$ на множестве X может быть сходящейся или расходящейся в зависимости от выбора метрики $\rho(\mathbf{x}, \mathbf{y})$. Последовательность $\{\mathbf{x}^{(k)}\}$ называется *фундаментальной*, если для любого числа $\varepsilon > 0$ найдется такой номер $N(\varepsilon)$, что $\rho(\mathbf{x}^{(k)}, \mathbf{x}^{(m)}) < \varepsilon$ при $k, m > N(\varepsilon)$. Если в метрическом пространстве X каждая фундаментальная последовательность сходится к некоторому пределу, являющемуся

элементом этого же пространства, то пространство X называется *полным*.

Пусть X и Y — два произвольных множества. Если каждому элементу $\mathbf{x} \in X$ поставлен в соответствие один и только один элемент $\mathbf{y} \in Y$, то говорят, что на X задано *отображение (оператор)* A множества X в Y и записывают $\mathbf{y} = A\mathbf{x}$.

Отображение A метрического пространства X в себя называется *сжимающим*, или *сжатием*, если существует такое число $0 < \alpha < 1$, что для любых двух точек $\mathbf{x}, \mathbf{y} \in X$ выполняется неравенство

$$\rho(A\mathbf{x}, A\mathbf{y}) \leq \alpha \rho(\mathbf{x}, \mathbf{y}). \quad (3.28)$$

Точка \mathbf{x} называется *неподвижной точкой отображения* A , если $A\mathbf{x} = \mathbf{x}$. Для уравнений вида $A\mathbf{x} = \mathbf{x}$ имеет место *теорема о неподвижной точке* (или *принцип сжатых отображений*), сформулированная польским математиком С. Банахом.

Теорема. Всякое сжимающее отображение, определенное в полном метрическом пространстве X , имеет одну и только одну неподвижную точку \mathbf{x}^* . Последовательность $\{\mathbf{x}^{(k)}\}$, определяемая равенством

$$\mathbf{x}^{(k+1)} = A\mathbf{x}^{(k)}, \quad k = 0, 1, 2, \dots, \quad (3.29)$$

сходится к точке x^* при любом выборе начального приближения $\mathbf{x}^{(0)} \in X$. Имеет место оценка

$$\rho(\mathbf{x}^*, \mathbf{x}^{(k)}) \leq \frac{\alpha^k}{1-\alpha} \rho(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}). \quad (3.30)$$

Доказательство. Пусть $\mathbf{x}^{(0)}$ — любая точка в X . Поскольку A есть отображение пространства X в себя, то все последовательные приближения $\mathbf{x}^{(k)} \in X$ ($k = 0, 1, 2, \dots$). Из условия сжатия отображения следует:

$$\begin{aligned} \rho(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) &= \rho(A\mathbf{x}^{(1)}, A\mathbf{x}^{(0)}) \leq \alpha \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}), \\ \rho(\mathbf{x}^{(3)}, \mathbf{x}^{(2)}) &= \rho(A\mathbf{x}^{(2)}, A\mathbf{x}^{(1)}) \leq \alpha \rho(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) \leq \alpha^2 \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}), \\ &\vdots \\ \rho(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) &\leq \alpha^k \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}). \end{aligned}$$

Покажем, что последовательность $\{\mathbf{x}^{(k)}\}$ — фундаментальная. Пусть $m > k$. Из неравенства треугольника вытекает:

$$\begin{aligned} \rho(\mathbf{x}^{(m)}, \mathbf{x}^{(k)}) &\leq \rho(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) + \rho(\mathbf{x}^{(k+2)}, \mathbf{x}^{(k+1)}) + \dots \\ &\dots + \rho(\mathbf{x}^{(m)}, \mathbf{x}^{(m-1)}) \leq (\alpha^k + \alpha^{k+1} + \dots + \alpha^{m-1}) \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}) = \\ &= \frac{\alpha^k (1 - \alpha^{m-k})}{1 - \alpha} \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}) < \frac{\alpha^k}{1 - \alpha} \rho(\mathbf{x}^{(1)}, \mathbf{x}^{(0)}). \quad (3.31) \end{aligned}$$

Так как $0 < \alpha < 1$, то для любого $\varepsilon > 0$ существует номер $N(\varepsilon)$ такой, что при $k > N(\varepsilon)$ и $m > N(\varepsilon)$ будет выполняться неравенство

$$\rho(\mathbf{x}^{(m)}, \mathbf{x}^{(k)}) < \varepsilon,$$

т. е. последовательность $\{\mathbf{x}^{(k)}\}$ фундаментальна. Из полноты пространства X следует существование предела

$$\mathbf{x}^* = \lim_{k \rightarrow \infty} \mathbf{x}^{(k)},$$

причем $\mathbf{x}^* \in X$.

Докажем, что \mathbf{x}^* — неподвижная точка отображения A . Из неравенства

$$\rho(A\mathbf{x}^{(k)}, A\mathbf{x}^*) \leq \alpha \rho(\mathbf{x}^{(k)}, \mathbf{x}^*) < \varepsilon$$

следует непрерывность сжимающего отображения. Поэтому

$$A\mathbf{x}^* = A \lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \lim_{k \rightarrow \infty} A\mathbf{x}^{(k)} = \lim_{k \rightarrow \infty} \mathbf{x}^{(k+1)} = \mathbf{x}^*.$$

Далее докажем единственность неподвижной точки. Предположим, что существует две неподвижные точки \mathbf{x}^* и $\mathbf{y}^* \in X$. Тогда $A\mathbf{x}^* = \mathbf{x}^*$, $A\mathbf{y}^* = \mathbf{y}^*$ и $\rho(\mathbf{x}^*, \mathbf{y}^*) = \rho(A\mathbf{x}^*, A\mathbf{y}^*) \leq \alpha \rho(\mathbf{x}^*, \mathbf{y}^*)$. Так как $0 < \alpha < 1$, то $\rho(\mathbf{x}^*, \mathbf{y}^*) = 0$, т. е. $\mathbf{x}^* = \mathbf{y}^*$.

Если в неравенстве (3.31) перейти к пределу при $m \rightarrow \infty$, то получим оценку (3.30).

Теорема доказана.

Занимаем неравенство (3.31):

$$\begin{aligned} \rho(\mathbf{x}^{(m)}, \mathbf{x}^{(k)}) &\leq \rho(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}) + \rho(\mathbf{x}^{(k+2)}, \mathbf{x}^{(k+1)}) + \dots \\ \dots + \rho(\mathbf{x}^{(m)}, \mathbf{x}^{(m-1)}) &\leq \alpha \rho(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}) + \alpha^2 \rho(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}) + \dots \\ \dots + \alpha^{m-k} \rho(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}) &= (\alpha + \alpha^2 + \dots + \alpha^{m-k}) \times \\ &\quad \times \rho(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}) = \frac{\alpha(1-\alpha^{m-k})}{1-\alpha} \rho(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}). \end{aligned}$$

При $m \rightarrow \infty$ получим

$$\rho(\mathbf{x}^*, \mathbf{x}^{(k)}) \leq \frac{\alpha}{1-\alpha} \rho(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}). \quad (3.32)$$

Метод простой итерации (метод Якоби). Преобразуем исходную систему алгебраических уравнений $A\mathbf{x} = \mathbf{b}$ к виду

$$\mathbf{x} = B\mathbf{x} + C. \quad (3.33)$$

Предположим, что выбрано начальное приближение $\mathbf{x}^{(0)} = (\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \dots, \mathbf{x}_n^{(0)})$ к точному решению \mathbf{x} . Обычно по-

лагают $\mathbf{x}^{(0)} = C$. Вычислим последовательные приближения в методе простой итерации

$$\mathbf{x}^{(k)} = B\mathbf{x}^{(k-1)} + C, \quad k = 1, 2, \dots \quad (3.34)$$

Итерационный процесс (3.34) называется *сходящимся* к решению \mathbf{x} системы (3.33), если при любом выборе начального приближения $\mathbf{x}^{(0)}$ выполнено условие

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}\| = 0.$$

Как было отмечено выше, из сходимости по норме следует также и покоординатная сходимость последовательных приближений.

Выясним условия сходимости метода простой итерации. Вычитая (3.33) из (3.34), получим

$$\mathbf{x}^{(k)} - \mathbf{x} = B(\mathbf{x}^{(k-1)} - \mathbf{x}). \quad (3.35)$$

Из этой рекуррентной формулы следует

$$\begin{aligned} \mathbf{x}^{(k)} - \mathbf{x} &= B(\mathbf{x}^{(k-1)} - \mathbf{x}) = B^2(\mathbf{x}^{(k-2)} - \mathbf{x}) = \dots \\ &\dots = B^k(\mathbf{x}^{(0)} - \mathbf{x}). \end{aligned}$$

Таким образом, вектор $\mathbf{x}^{(k)} \rightarrow \mathbf{x}$ при $k \rightarrow \infty$ тогда и только тогда, если степень матрицы B^k стремится к нулевой матрице Θ при $k \rightarrow \infty$.

Известно [21], что для любой квадратной матрицы B матрица $B^k \rightarrow \Theta$ при $k \rightarrow \infty$ тогда и только тогда, когда все ее собственные числа по модулю меньше единицы. Из этого утверждения следует необходимое и достаточное условие сходимости метода простой итерации.

Теорема. Пусть система (3.33) имеет решение. Метод простой итерации (3.34) сходится при любом начальном приближении $\mathbf{x}^{(0)}$ к решению \mathbf{x} тогда и только тогда, если все собственные числа матрицы B по модулю меньше единицы.

На практике таким критерием трудно воспользоваться, так как задача определения собственных чисел более сложная, чем решение линейной системы. Удобнее в качестве критерия сходимости использовать норму матрицы. Учитывая, что собственные числа матрицы B и ее норма связаны неравенством (3.21)

$$|\lambda| \leq \|B\|,$$

сформулируем достаточные условия сходимости: метод простой итерации (3.34) сходится к решению системы (3.33), если любая согласованная норма матрицы B меньше единицы,

$$\|B\| < 1. \quad (3.36)$$

Если выбрать в пространстве R^n норму вектора $\|\mathbf{x}\|$ и ввести метрику $\rho(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$, то отображение (3.33) будет сжатием, если любая согласованная норма матрицы B меньше единицы, т. е. условие (3.36) следует из принципа сжатых отображений.

Учитывая выражения (3.17)–(3.20) для матричных норм, достаточные условия сходимости метода простой итерации (3.34) можно представить так:

$$\begin{aligned}\|B\|_1 &= \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}| < 1, \\ \|B\|_\infty &= \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}| < 1, \\ \|B\|_E &= \left(\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2 \right)^{1/2} < 1.\end{aligned}\quad (3.37)$$

Если исходную систему алгебраических уравнений $A\mathbf{x} = \mathbf{b}$ свести к виду (3.33):

$$x_i = - \sum_{\substack{j=1, \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j + \frac{b_i}{a_{ii}}, \quad a_{ii} \neq 0, \quad i = 1, 2, \dots, n,$$

то при наличии в матрице A диагонального преобладания

$$\sum_{\substack{j=1, \\ j \neq i}}^n |a_{ij}| < |a_{ii}|, \quad i = 1, 2, \dots, n, \quad (3.38)$$

выполняется условие сходимости метода простой итерации, так как $\|B\|_\infty < 1$.

Для погрешности метода простой итерации согласно (3.32) получим оценку

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|B\|}{1 - \|B\|} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|. \quad (3.39)$$

Метод сходится со скоростью геометрической прогрессии, знаменатель которой равен $\|B\|$.

Для достижения заданной точности ϵ , т. е. выполнения неравенств

$$|x_i - x_i^{(k)}| < \epsilon, \quad i = 1, 2, \dots, n,$$

итерации обычно продолжают до тех пор, пока не выполняются условия

$$|x_i^{(k)} - x_i^{(k-1)}| \leq \frac{1 - \|B\|}{\|B\|} \epsilon, \quad i = 1, 2, \dots, n. \quad (3.40)$$

Для характеристики скорости сходимости итерационных методов вводится понятие порядка метода [2]. Считают, что метод имеет p -й порядок, если существуют $C_1 > 0$ и $C_2 < \infty$ такие, что

$$\rho(\mathbf{x}^{(k+1)}, \mathbf{x}) \leq C_2 (\rho(\mathbf{x}^{(k)}, \mathbf{x}))^\rho$$

при условии $\rho(\mathbf{x}^{(k)}, \mathbf{x}) \leq C_1$. При малых значениях C_1 итерационный процесс сходится тем быстрее, чем больше ρ . В частности, метод простой итерации имеет первый порядок, а рассматриваемый ниже метод Ньютона — второй порядок.

Метод Зейделя. В итерационном методе Зейделя последовательно уточняются компоненты решения системы (3.33) следующим образом: при вычислении $(k+1)$ -го приближенного значения x_i при $i > 1$ используются уже вычисленные ранее $(k+1)$ -е приближения неизвестных $x_1, x_2 \dots, x_{i-1}$ и (k) -е приближения неизвестных x_i, x_{i+1}, \dots, x_n .

$$\left\{ \begin{array}{l} x_1^{(k+1)} = c_1 + b_{11}x_1^{(k)} + b_{12}x_2^{(k)} + \cdots + b_{1n}x_n^{(k)}, \\ \vdots \\ x_i^{(k+1)} = c_i + b_{i1}x_1^{(k+1)} + \cdots + b_{i, i-1}x_{i-1}^{(k+1)} + \\ \quad + b_{ii}x_i^{(k)} + \cdots + b_{in}x_n^{(k)}, \\ \vdots \\ x_n^{(k+1)} = c_n + b_{n1}x_1^{(k+1)} + \cdots + b_{ni}x_i^{(k+1)} + \cdots \\ \quad + b_{n, n-1}x_{n-1}^{(k+1)} + b_{nn}x_n^{(k)}, \quad k = 0, 1, 2, \dots \end{array} \right. \quad (3.41)$$

Систему (3.33) запишем в матричной форме:

$$x^{(k+1)} = C + B_1 X^{(k+1)} + B_2 X^{(k)}, \quad k = 0, 1, 2, \dots, \quad (3.42)$$

где

$$B_1 = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ b_{21} & 0 & \dots & 0 & 0 \\ b_{31} & b_{32} & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ b_{n1} & b_{n2} & \dots & b_{n,n-1} & 0 \end{pmatrix}, \quad B_2 = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ 0 & b_{22} & \dots & b_{2n} \\ 0 & 0 & \dots & b_{3n} \\ \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & \dots & b_{nn} \end{pmatrix},$$

$B_1 + B_2 = B.$

Метод Зейделя (3.42) эквивалентен методу простой итерации для системы

$$\mathbf{x} = C_1 + B_3 \mathbf{x},$$

где

$$B_3 = (E - B_1)^{-1} B_2, \quad C_1 = (E - B_1)^{-1} C.$$

Поэтому условия сходимости процесса итерации легко перенормулировать для данного случая: метод Зейделя сходится,

если любая норма матрицы B_3 меньше единицы. Области сходимости методов простой итерации и Зейделя не совпадают, но пересекаются. В частности, при выполнении условия (3.38) метод Зейделя сходится. Обычно метод Зейделя дает более быструю сходимость, чем метод простой итерации, хотя это бывает не всегда.

В практических вычислениях итерационный процесс прекращают, если два последовательных приближения отличаются меньше наперед заданного ϵ в смысле выбранной нормы,

$$\| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \| < \epsilon. \quad (3.43)$$

Решение системы алгебраических уравнений (3.33) итерационным методом Зейделя реализуется программой 5 (при $n \leq 9$). Последовательные приближения записываются на каждом шаге и сравниваются с предыдущими.

Программа 5 решения системы n ($n \leq 9$) линейных алгебраических уравнений итерационным методом Зейделя по формулам (3.41).

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИПД	6Г	10	КИП0	Г0	20	ИП0	60
01	П1	41	11	С/П	50	21	ПС	4С
02	F,	25	12	Х	12	22	F,	25
03	С/П	50	13	+	10	23	FL2	58
04	ИПД	6-	14	ИП2	62	24	10	10
05	П2	42	15	ИП1	61	25	КПС	LС
06	3	03	16	-	11	26	FL1	5L
07	+	10	17	FX = 0	5Е	27	03	03
08	П0	40	18	22	22	28	БП	51
09	XУ	14	19	F,	25	29	00	00

Распределение регистров памяти:

Номер регистра	0, С	1, 2	3, ..., В	Д
Переменная	Адрес x_i	Счетчики циклов	x_n, \dots, x_1	n

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В режиме автоматической работы (F АВТ) ввести начальное приближение решения и количество уравнений:

$$(x_n^{(0)}) \text{ П3 } (x_{n-1}^{(0)}) \text{ П4 } \dots (x_1^{(0)}) \text{ П } (n+2) \text{ (n) ПД}$$

3. Нажать клавиши В/О С/П. Приступить к вводу коэффициентов и вычислениям по схеме:

$$(c_1) \text{С/П } (b_{11}) \text{ С/П } \dots (b_{1n}) \text{ С/П } (\text{прочесть } x_1^{(0)}) \\ (c_2) \text{ С/П } (b_{21}) \text{ С/П } \dots (b_{2n}) \text{ С/П } (\text{прочесть } x_2^{(0)}) \dots \\ (c_n) \text{ С/П } (b_{n1}) \text{ С/П } \dots (b_{nn}) \text{ С/П } (\text{прочесть } x_n^{(0)}).$$

Для нахождения каждого следующего приближения повторяем вычисления по той же схеме.

Контрольный пример.

Решить систему:

$$\begin{cases} x_1 = 1,99 + 0 \cdot x_1 - 0,243x_2 - 0,168x_3, \\ x_2 = 2,94 - 0,304x_1 + 0 \cdot x_2 - 0,212x_3, \\ x_3 = 2,53 + 0,126x_1 + 0,172x_2 + 0 \cdot x_3 \end{cases}$$

методом Зейделя с начальным приближением $x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 2$.

Итерации продолжать до совпадения трех десятичных знаков.

Результаты вычислений заносим в таблицу (k — номер итерации).

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
1	1,168	2,161	3,049
2	0,953	2,004	2,995
3	1,000	2,001	3,000
4	1,000	2,000	3,000

Ответ: $x_1 = 1,000$; $x_2 = 2,000$; $x_3 = 3,000$.

Глава 4. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ И СИСТЕМ

4.1. Численные методы решения нелинейного уравнения с одним неизвестным

Пусть дано уравнение

$$f(x) = 0, \quad (4.1)$$

где $f(x)$ — алгебраическая или трансцендентная функция

с одним неизвестным. Если $f(x^*) = 0$, то x^* называется *корнем уравнения* (4.1) или *нулем функции* $f(x)$. Будем предполагать, что уравнение (4.1) имеет лишь изолированные корни, т. е. для каждого корня существует окрестность, не содержащая других корней этого уравнения.

Приближенное вычисление изолированных действительных корней уравнения (4.1) состоит из двух этапов:

1) отделения корней — выделения отрезка, принадлежащего области существования функции $f(x)$, на котором расположен один и только один корень;

2) уточнения приближенных корней, т. е. вычисления их с требуемой точностью.

Процесс отделения корней уравнения (4.1) основан на теореме Больцано—Коши: если непрерывная функция $f(x)$ принимает на концах отрезка $[a; b]$ значения разных знаков, т. е. $f(a) \cdot f(b) < 0$, то внутри этого отрезка содержится по крайней мере один корень. Этот корень будет единственным, если производная $f'(x)$ существует и сохраняет постоянный знак внутри интервала $[a; b]$.

Замечание. Отделение корней уравнения (4.1) можно выполнить графически, если удается построить график функции и найти приближенно точки его пересечения с осью абсцисс. Иногда проще заменить уравнение (4.1) эквивалентным ему уравнением $\varphi(x) = g(x)$, а затем найти приближенно абсциссы точек пересечения графиков функций $y = \varphi(x)$ и $y = g(x)$.

Рассмотрим методы уточнения приближенных корней.

Метод половинного деления, или метод биссекции, является самым простым и надежным алгоритмом нахождения корней уравнения (4.1). Пусть функция $f(x)$ непрерывна на отрезке $[a; b]$, на концах его принимает значения разных знаков, т. е. $f(a) \cdot f(b) < 0$, и производная $f'(x)$ сохраняет на этом отрезке знак. Требуется найти приближенное значение корня уравнения (4.1), принадлежащего отрезку $[a; b]$, с точностью ε . В методе половинного деления отрезок $[a; b]$

делим пополам. Если $f\left(\frac{a+b}{2}\right) = 0$, то полагаем $x^* = \frac{a+b}{2}$.

В противном случае обозначим через $[a_1; b_1]$ ту часть отрезка, на концах которой функция принимает значения разных знаков. Процесс последовательного деления продолжаем до тех пор, пока не будет выполняться одно из условий:

$$1) f\left(\frac{a_n + b_n}{2}\right) = 0, \text{ тогда } x^* = \frac{a_n + b_n}{2};$$

2) при некотором n длина отрезка, содержащего корень,

станет меньше заданного 2ϵ , т. е. $|b_n - a_n| < 2\epsilon$, тогда $x^* = \frac{a_n + b_n}{2}$.

Количество итераций, необходимое для достижения точности ϵ , можно оценить заранее:

$$\frac{b-a}{2^n} < 2\epsilon, \text{ отсюда } n > \frac{\ln \left(\frac{b-a}{2\epsilon} \right)}{\ln 2}.$$

Метод устойчив к ошибкам округления. Но сходится он медленно. При увеличении точности значительно возрастает объем вычислительной работы. На практике метод часто используют для грубого определения начального приближения к корню, а затем применяют быстрее сходящийся итерационный метод.

Описание алгоритма.

1. Ввести исходные данные a, b, ϵ .
2. Вычислить $f(a)$.
3. Вычислить $x = \frac{a+b}{2}, f(x)$.
4. Если $f(x) \neq 0$, то перейти к п. 5, иначе к п. 7.
5. Если $x - a \geq \epsilon$, перейти к п. 6, иначе к п. 7.
6. Если $f(a) \cdot f(x) > 0$, положить $a = x, f(a) = f(x)$, иначе положить $b = x$. Перейти к п. 3.
7. Вывести значения x и $f(x)$. Конец.

Программа 6 вычисления корня уравнения $f(x) = 0$ на отрезке $[a; b]$, где $f(a) f(b) < 0$, методом половинного деления с точностью ϵ в автоматическом режиме вычислений.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИПА	6—	14	П2	42	28	04	04
01	ПП	53	15	FX ≠ 0	57	29	ИПВ	6L
02	39	39	16	35	35	30	ПА	4—
03	П1	41	17	ИПВ	6L	31	ИП2	62
04	ИПВ	6L	18	ИПА	6—	32	П1	41
05	ПС	4C	19	ИП0	60	33	БП	51
06	ИПА	6—	20	+	10	34	06	06
07	ИПС	6C	21	—	11	35	ИПВ	6L
08	+	10	22	FX ≥ 0	59	36	С/П	50
09	2	02	23	35	35	37	ИП2	62
10	÷	13	24	ИП1	61	38	С/П	50
11	ПВ	4L	25	ИП2	62	39
12	ПП	53	26	×	12
13	39	39	27	FX ≥ 0	59	...	В/0	52

Распределение регистров памяти:

Номер регистра	0	A	B	C	I	2	3, ..., 9
Переменная	ε	a	b	b, x	$f(a)$	$f(x)$	свободны

Инструкция.

- Составить подпрограмму вычисления значений функции $f(x)$. Значение аргумента содержится в РХ.
- В режиме программирования (F ПРГ) ввести программу и подпрограмму.
- В автоматическом режиме (F АВТ) ввести исходные данные: (ε) ПО (a) ПА (b) ПВ.
- Нажать клавиши В/О С/П. После ОСТАНОВа записать с индикатора приближенное значение корня x^* . Нажать клавиши ИП2 и записать значение $f(x^*)$.
- Для нового счета составить подпрограмму вычисления значений новой функции, ввести ее с адреса 39 и перейти к п. 3.

Контрольный пример. Найти корень уравнения $x^3 - 2x - 5 = 0$, на отрезке $[2; 3]$ с точностью $\varepsilon = 10^{-3}$.

Подпрограмма вычисления значений функции $f(x)$:
 $\uparrow FX^2 2 - \times 5 - B/O$.

В результате вычислений получим $x^* = 2,0947265$, где три цифры после запятой точные.

Ответ: $f(x^*) = -0,0019536$, $x^* = 2,095$.

Время вычислений 1 мин 20 с.

Метод простой итерации, или метод последовательных приближений, является одним из самых общих методов решения уравнения (4.1). Заменим уравнение $f(x) = 0$ эквивалентным ему уравнением

$$x = \varphi(x). \quad (4.2)$$

Это можно сделать многими способами, положив, например, $\varphi(x) = x + g(x) \cdot f(x)$, где $g(x)$ — произвольная непрерывная знакопостоянная функция. Выберем некоторое нулевое приближение $x_0 \in [a; b]$ корня уравнения (4.2) и подставим его в правую часть. Получим $x_1 = \varphi(x_0)$. Дальнейшие приближения вычислим по формулам

$$x_{n+1} = \varphi(x_n) \quad (n = 1, 2, \dots). \quad (4.3)$$

Если последовательность $\{x_n\}$ сходится, т. е. существует предел $\lim_{n \rightarrow \infty} x_n = x^*$, то переходя к пределу в (4.3), получим для непрерывной функции

$$\lim_{n \rightarrow \infty} x_{n+1} = \varphi(\lim_{n \rightarrow \infty} x_n), \text{ или } x^* = \varphi(x^*).$$

Таким образом, предел x^* является корнем уравнения (4.2). Решение x^* уравнения (4.2) является неподвижной точкой отображения $\varphi(x)$. Условия сходимости итерационного процесса (4.3) получим из принципа сжатых отображений (см. п. 3.6). Пусть функция $\varphi(x)$ дифференцируема на $[a; b]$, причем все ее значения принадлежат отрезку $[a; b]$. Если ввести расстояние $\rho(x, y) = |x - y|$, то множество точек отрезка $[a; b]$ будет полным метрическим пространством.

Используя формулу Лагранжа, получим:

$$|\varphi(x) - \varphi(y)| = |\varphi'(c)| |x - y| \leq q \cdot \rho(x, y),$$

где точка c лежит между точками x и y , x и $y \in [a; b]$.

Таким образом, если выполнено условие

$$|\varphi'(x)| \leq q < 1 \text{ для всех } x \in [a; b], \quad (4.4)$$

то отображение будет сжимающим и процесс итерации (4.3) сходится к единственному корню уравнения (4.2) на отрезке $[a; b]$ независимо от выбора начального приближения $x_0 \in [a; b]$.

Очевидно, что чем меньше q , тем быстрей сходимость. На основании (3.32) получим оценку погрешности метода простой итерации (4.3):

$$|x^* - x_n| \leq \frac{q}{1-q} |x_n - x_{n-1}| \quad (4.5)$$

$$(n = 1, 2, 3, \dots).$$

При $0 < \varphi'(x) \leq q < 1$ процесс итерации можно прекратить, если для двух последовательных приближений обеспечено выполнение неравенства

$$|x_n - x_{n-1}| \leq \frac{1-q}{q} \varepsilon, \quad (4.6)$$

где ε — заданная точность. В частности, при $0 < \varphi'(x) \leq \frac{1}{2}$ и при $-1 < \varphi'(x) < 0$ x_n будет приближенным значением корня x^* с точностью ε , т. е. $|x^* - x_n| \leq \varepsilon$, если $|x_n - x_{n-1}| \leq \varepsilon$. При $\frac{1}{2} < \varphi'(x) < 1$ выполнение условия $|x_n - x_{n-1}| \leq \varepsilon$ может оказаться недостаточным для получения приближения с точностью ε . В этом случае итерации прекращают, если для трех последовательных приближений выполняется неравенство [19]:

$$\left| \frac{q}{1-q} (x_n - x_{n-1}) \right| = \frac{(x_n - x_{n-1})^2}{|2x_{n-1} - x_n - x_{n-2}|} \leq \varepsilon. \quad (4.7)$$

Если $|\varphi'(x)|$ трудно оценить, то для оценки погрешности удобно использовать неравенство (4.7).

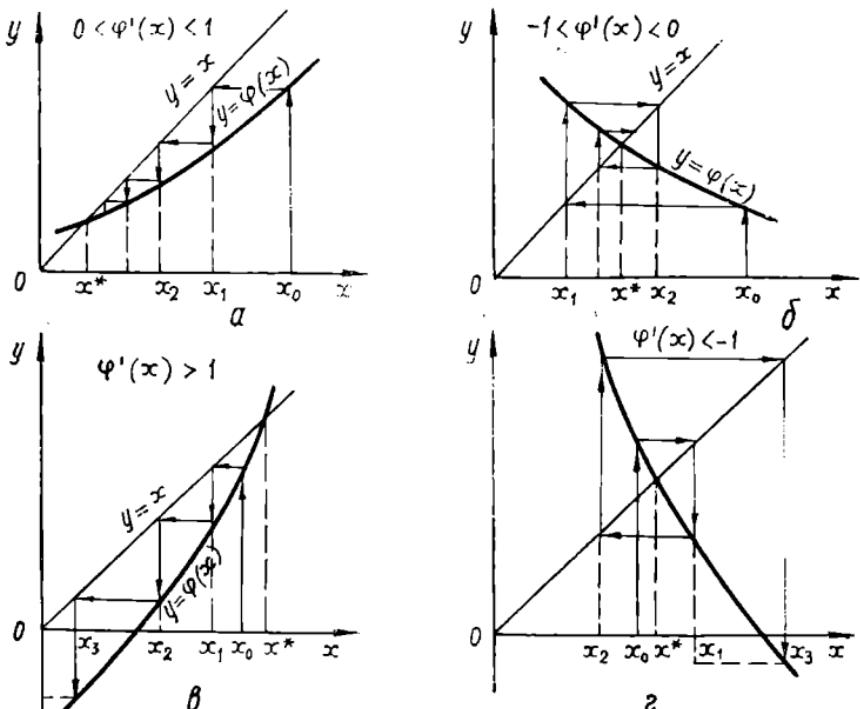


Рис. 8

Геометрическая интерпретация метода итераций для случаев $0 < \varphi'(x) < 1$, $-1 < \varphi'(x) < 0$, $\varphi'(x) > 1$ и $\varphi'(x) < -1$ представлена на рис. 8, а—г.

Замечание. В уравнении (4.2) функцию $\varphi(x)$ нужно выбрать так, чтобы выполнялось условие (4.4). Например, если для решения уравнения $x^2 = a$ выбрать итерационный процесс $x_{n+1} = \frac{a}{x_n}$, то он будет расходящимся, в то время как последовательные приближения $x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right)$ сходятся очень быстро при любом $x_0 > 0$. Для уравнения $x = 4 - 2^x$ процесс итераций (4.3) на отрезке $[1; 2]$ расходится, но если уравнение преобразовать к виду $x = \log_2(4-x)$, то процесс сходится.

На практике итерирующую функцию часто выбирают в виде $\varphi(x) = x - \alpha f(x)$. При этом выполнение условия (4.4) сходимости итерационного процесса обеспечивается выбором постоянного числа α , удовлетворяющего условию:

$$|\varphi'(x)| = |1 - \alpha f'(x)| < 1, \text{ или } 0 < \alpha f'(x) < 2,$$

откуда $\alpha f'(x) > 0$ и $|\alpha| \leq \frac{2}{m}$, где $|f'(x)| \geq m$.

Программа 7 решения уравнения $x = \varphi(x)$ методом простой итерации по формулам

$$x_{n+1} = \varphi(x_n) \quad (n = 0, 1, 2, \dots)$$

в полуавтоматическом режиме:

ПО С/П БП 00
Программа вычисления $\varphi(x)$

При написании программы вычисления $\varphi(x)$ следует учесть, что значение x хранится в РХ и Р0. После ввода программы вводим начальное приближение x_0 и осуществляем пуск: (х₀) В/О С/П. После ОСТАНОВа спишем значение x_1 . Для вычислений последующих приближений x_n достаточно нажимать клавишу С/П.

Контрольный пример. Найти приближенное решение уравнения $x + \ln x = 5$ на отрезке $[3; 4]$ с погрешностью, не превышающей $\varepsilon = 10^{-4}$.

Запишем уравнение в виде $x = 5 - \ln x$, тогда $\varphi(x) = 5 - \ln x$, $\varphi'(x) = -\frac{1}{x}$, причем $-1 < -\frac{1}{x} < 0$, если $x \in [3; 4]$, т. е. выполняется условие сходимости итерационного процесса. В результате вычислений получим последовательные приближения корня

x_0	x_1	x_2	x_3	x_4
4	3,6137056	3,7152661	3,6875498	3,6950378

x_5	x_6	x_7	x_8
3,6930092	3,6935585	3,6934098	3,6934499

В последующих итерациях четыре цифры после запятой совпадают.

Omværm: $x = 3,6934$.

Метод Ньютона—Рафсона, или метод касательных, является наиболее часто употребляемым методом уточнения корней, пригодным для решения алгебраических и трансцендентных уравнений.

Пусть в уравнении $f(x) = 0$ функция $f(x)$ дважды непрерывно дифференцируема на отрезке $[a; b]$, содержащем искомый отделенный корень x^* , причем производные отличны от нуля, знакопостоянны и начальное приближение корня x_0 принадлежит отрезку $[a; b]$. Геометрически это означает, что график функции $y = f(x)$ в любой точке отрезка $[a; b]$ имеет

касательную и не имеет на отрезке экстремумов и точек перегиба. Выберем на кривой произвольную точку $M_0(x_0, f(x_0))$, $x_0 \in [a; b]$ и проведем касательную к кривой в этой точке. Уравнение касательной:

$$y - f(x_0) = f'(x_0)(x - x_0).$$

Найдем точку x_1 пересечения касательной с осью абсцисс. Учитывая, что $y_1 = 0$, получим $-f(x_0) = f'(x_0)(x_1 - x_0)$, отсюда получаем первое приближение корня:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Через точку $M_1(x_1, f(x_1))$ снова проведем касательную к кривой, точка пересечения которой с осью Ox даст нам второе приближение корня, и т. д.

Продолжив описанный процесс построения касательных и вычисления точек их пересечения с осью Ox , получим формулу итерационного метода Ньютона—Рафсона:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (4.8)$$

$$(n = 0, 1, 2, \dots).$$

Геометрически метод Ньютона—Рафсона означает замену дуги кривой на каждой итерации касательной к ней в точке x_n (рис. 9).

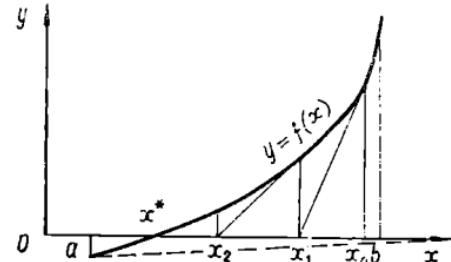


Рис. 9

Метод можно рассматривать как частный случай метода простых итераций (4.3) для функции $\varphi(x) = x - f(x)/f'(x)$, тогда $\varphi'(x) = f(x) \cdot f''(x) / (f'(x))^2$. Поскольку $\varphi'(x^*) = 0$, то ввиду непрерывности $\varphi'(x)$ вблизи корня x^* выполняется условие сходимости итерационного процесса (4.4).

Таким образом, если нулевое приближение выбрано достаточно близко к корню, то метод Ньютона всегда сходится. При произвольном нулевом приближении метод (4.8) сходится, если для всех $x \in [a; b]$ выполнено условие

$$\frac{|f(x) \cdot f''(x)|}{(f'(x))^2} < 1. \quad (4.9)$$

Отметим одно важное достаточное условие сходимости метода Ньютона—Рафсона. Пусть $f(a) \cdot f(b) < 0$, а $f'(x)$ и $f''(x)$ отличны от нуля и сохраняют определенные знаки на

отрезке $[a; b]$. Тогда для любого $x_0 \in [a; b]$, удовлетворяющего условию

$$f(x_0) \cdot f''(x_0) > 0, \quad (4.10)$$

метод сходится [14]. При выполнении условия (4.10) приближение x_{n+1} будет лежать между x_n и x^* , т. е. ближе к корню, поэтому при возрастании n приближения x_n монотонно стремятся к точному решению. Неудачный выбор начального приближения ($x_0 = a$ на рис. 9) может привести к тому, что последующие приближения выйдут за пределы отрезка $[a; b]$.

Близкое к корню начальное приближение метода Ньютона—Рафсона можно получить с помощью другого метода, например метода половинного деления.

Оценим скорость сходимости метода вблизи простого корня. Воспользуемся формулой Тейлора

$$0 = f(x^*) = f(x_n) + f'(x_n)(x^* - x_n) + \frac{1}{2} f''(c)(x^* - x_n)^2,$$

где c лежит между x_n и x^* . Отсюда

$$x^* - x_n + \frac{f(x_n)}{f'(x_n)} = -\frac{1}{2} \frac{f''(c)}{f'(x_n)} (x^* - x_n)^2. \quad (4.11)$$

Но на основании (4.8)

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

поэтому

$$x^* - x_{n+1} = -\frac{1}{2} \frac{f''(c)}{f'(x_n)} (x^* - x_n)^2. \quad (4.12)$$

Обозначим через M наибольшее значение $|f''(x)|$, а через m — наименьшее значение $|f'(x)|$ на отрезке $[a; b]$. Тогда абсолютные погрешности двух последовательных приближений x_n и x_{n+1} связаны неравенством

$$|x^* - x_{n+1}| \leq \frac{M}{2m} (x^* - x_n)^2. \quad (4.13)$$

Оценка (4.13) показывает, что погрешность каждого нового приближения уменьшается пропорционально квадрату погрешности предыдущего. Поэтому, если начальное приближение x_0 выбрать так, чтобы

$$\frac{M}{2m} |x^* - x_0| < 1,$$

то каждое новое приближение удваивает число верных десятичных знаков, т. е. сходимость будет квадратичной [14].

Метод Ньютона—Рафсона обеспечивает быструю сходимость, если известно хорошее начальное приближение корня. Если два последовательные приближения x_{n+1} и x_n , полу-

ченные по методу Ньютона—Рафсона, совпадают с точностью ϵ , то это еще не гарантирует совпадения с той же точностью приближения x_{n+1} с точным решением x^* . Для оценки погрешности можно воспользоваться следующим неравенством:

$$|x^* - x_{n+1}| \leq \frac{M}{2m} |x_{n+1} - x_n|^2. \quad (4.14)$$

Таким образом, для определения корня уравнения (4.1) по методу Ньютона—Рафсона с точностью ϵ итерационный процесс продолжаем до тех пор, пока не получим

$$|x_{n+1} - x_n| \leq \sqrt{2m\epsilon/M}. \quad (4.15)$$

Если x^* — кратный корень уравнения (4.1), так что $f'(x^*) = 0$, то метод Ньютона—Рафсона имеет линейную сходимость, т. е. сходится со скоростью геометрической прогрессии [19].

Кратные корни можно находить так же, как корни уравнения

$$f'(x) = 0,$$

или

$$f^{(n)}(x) = 0, \quad n = 1, 2, \dots, k-1,$$

где k — кратность корня.

Замечание. Если производная $f'(x)$ мало изменяется на $[a; b]$ и ее вычисление громоздко, то итерации можно проводить по формуле:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} \quad (n = 0, 1, 2, \dots). \quad (4.16)$$

Итерационную схему (4.16) называют модифицированным методом Ньютона.

Метод секущих. Если вычисление производной $f'(x)$ затруднительно по каким-либо причинам, то более удобно применение метода секущих. Здесь производная $f'(x_n)$ заменяется первой разделиенной разностью, найденной по двум последним итерациям:

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

Подставив выражение $f'(x_n)$ в (4.8), получим итерационный процесс метода секущих:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \quad (n = 1, 2, \dots). \quad (4.17)$$

Для начала процесса нужно задать x_0 и x_1 . На каждом шаге метода секущих участок кривой заменяется секущей, проходящей через точки с абсциссами x_n и x_{n-1} (рис. 10).

Установлено, что если x^* — корень уравнения $f(x) = 0$, а $f'(x^*) \neq 0$, $f''(x^*) \neq 0$ и $f''(x)$ — непрерывная функция, то существует такая окрестность точки x^* , что если x_0 и x_1 — различные точки этой окрестности, то метод секущих (4.17) сходится к x^* . При этом [19]

$$|x_{n+1} - x^*| \leq C |x_n - x^*|^p,$$

где $C \neq 0$ — const, $p \approx 1,6$.

Таким образом, метод секущих сходится медленнее метода Ньютона, однако на каждой итерации требует вычисления только функции. Следует иметь в виду, что вблизи корня x^* , особенно если он кратный, значения $f(x_n)$ и $f'(x_{n-1})$ малы и близки, при делении на их разность в методе возникает потеря значащих цифр. Поэтому итерации в методе секущих проводят до тех пор, пока разности $|x_{n+1} - x_n|$ практически перестанут убывать.

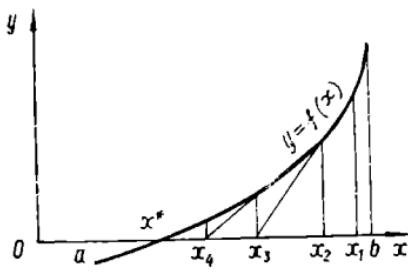


Рис. 10

4.2. Лабораторная работа 2. Приближенное решение нелинейного уравнения методом Ньютона—Рафсона

Изложенный выше метод Ньютона—Рафсона реализуется на микрокалькуляторе в виде программы 8.

Программа 8 приближенного решения уравнения $f(x) = 0$ методом Ньютона—Рафсона по формулам (4.8):

Адрес	Команда	Код	Содержание операции
00	ПП	53	
01	(α)	α	} Вызов подпрограммы вычисления $f'(x_n)$
02	П2	42	Занесение $f'(x_n)$ в Р2
03	ПП	53	
04	13	13	} Вызов подпрограммы вычисления $f(x_n)$
05	П1	41	Занесение $f(x_n)$ в Р1
06	ИП2	62	
07	\div	13	
08	ИП0	60	
09	ХУ	14	} Вычисление $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
10	—	11	
11	П0	40	
12	С/П	50	Индикация значения x_{n+1}
13	
...	} Подпрограмма вычисления $f(x_n)$
...	В/0	52	
α	
...	
...	В/0	52	

Распределение регистров памяти:

Номер регистра	0	1	2
Переменная	x_n	$f(x_n)$	$f'(x_n)$

Инструкция. 1. Составить подпрограммы вычисления функции $f(x)$ и ее производной $f'(x)$. Значение аргумента содержится в Р0.

2. В режиме программирования (F ПРГ) ввести программу и подпрограммы. Адрес α отвечает адресу начала подпрограммы вычисления производной $f'(x)$.

3. В автоматическом режиме (F АВТ) занести начальное приближение x_0 : (x_0) ПО.

4. Вычислить последующие приближения корня, нажимая клавиши В/О С/П.

После ОСТАНОВов списать с индикатора значения x_1 , x_2 и т. д. При необходимости значение функции $f(x_n)$ можно вызвать из Р1.

5. Вычисления продолжать до совпадения n знаков после запятой в двух последовательных приближениях. Для оценки погрешности можно воспользоваться формулой (4.14).

Если погрешность ϵ задана наперед, то можно воспользоваться неравенством (4.15).

Контрольный пример. Найти корень уравнения $x^3 - x^2 + 1 = 0$ на отрезке $[-1; -0,5]$. Итерации проводить до совпадения четырех знаков после запятой. Оценить погрешность.

Имеем: $f(x) = x^3 - x^2 + 1$, $f'(x) = 3x^2 - 2x$, $f''(x) = 6x - 2$, $m = \min_{[-1; -0,5]} |3x^2 - 2x| > 1$, $M = \max_{[-1; -0,5]} |6x - 2| = 8$.

Если $|x_{n+1} - x_n| \leq 10^{-4}$, то $|x_{n+1} - x^*| \leq \frac{8}{2} \cdot 10^{-8} < 10^{-7}$. Поскольку $f(-1)f'(-1) > 0$, то полагаем $x_0 = -1$. Составляем подпрограммы вычисления $f(x)$ и $f'(x)$, начиная с 13-го адреса:

ИПО ↑ FX² × BX — 1 + В/О ИПО
3 × 2 — ИПО × В/О

Значение α соответствует адресу 22. Последовательные приближения удобно заносить в таблицу:

n	0	1	2	3	4
x_n	-1	-0,8	-0,75681818	-0,75488146	-0,75487767

После четырех итераций вычисления прекращаем. Ответ: $x = -0,7548777 \pm 0,0000001$, $f(x) = 0$.

Задания к лабораторной работе 2. Найти корень уравнения на отрезке $[a; b]$ методом Ньютона—Рафсона. Вычисления проводить до

совпадения четырех знаков после запятой. Оценить погрешность. Переключатель Р—Г на панели поставить в положение Р.

- 1) $e^x + x = 0$, $[-1; 0]$;
- 2) $e^{-x} - \ln x = 0$, $[1; 2]$;
- 3) $\sin x - \frac{1}{x} = 0$, $[1; 1,5]$;
- 4) $\cos x - \frac{1}{x+2} = 0$, $[-1; 0]$;
- 5) $\cos x + \frac{1}{x+2} = 0$, $[1; 2]$;
- 6) $x^3 + x - 3 = 0$, $[1; 2]$;
- 7) $x^3 + x^2 - 3 = 0$, $[1; 2]$;
- 8) $e^{-x} - x = 0$, $[0; 1]$;
- 9) $\cos x + \frac{1}{x-2} = 0$, $[0; 1]$;
- 10) $\cos x - \frac{1}{x-2} = 0$, $[-2; -1]$;
- 11) $x^3 - x^2 + 3 = 0$, $[-2; -1]$;
- 12) $\ln x + x = 0$, $[0,4; 1]$;
- 13) $x^3 + x + 3 = 0$, $[-2; -1]$;
- 14) $\lg x + x = 0$, $[0,2; 1]$;
- 15) $x^2 - \cos x = 0$, $[-0,8; -0,7]$;
- 16) $x^3 + 3x^2 - 3 = 0$, $[-3, -2,2]$;
- 17) $x^2 - \cos x = 0$, $[0,7; 0,8]$;
- 18) $x^3 - 3x - 1 = 0$, $[-2; -1]$;
- 19) $\cos(x-1,1) - 3x + 2 = 0$, $[0,9; 1,1]$;
- 20) $x^2 + \sin 2x - 2 = 0$, $[-1,5; -1,4]$;
- 21) $x^3 + 6x^2 + 9x + 1 = 0$, $[-1; 0]$;
- 22) $4x^2 - \cos x - 4 = 0$, $[1; 1,2]$;
- 23) $2x^3 + 2x - 1 = 0$, $[0; 1]$;
- 24) $x^3 - 3x^2 + 1 = 0$, $[-1; 0]$;
- 25) $x^3 + x^2 + 3 = 0$, $[-2; -1]$;
- 26) $e^x + 2 \sin x = 0$, $[-1; 0]$;
- 27) $x - \frac{1}{x^2 + 1} = 0$, $[0; 1]$;
- 28) $x + \frac{1}{x^2 + 1} = 0$, $[-1; 0]$;
- 29) $x^3 + x^2 + 2 = 0$, $[-2; -1]$;
- 30) $x^3 - x^2 - 2 = 0$, $[1; 2]$.

4.3. Численные методы решения системы нелинейных уравнений

Рассмотрим систему n нелинейных уравнений с n неизвестными

$$f_i(x_1, x_2, \dots, x_n) = 0 \quad (i = 1, 2, \dots, n), \quad (4.18)$$

где f_i — некоторые алгебраические или трансцендентные функции. Обозначив $\mathbf{x} = (x_1, x_2, \dots, x_n)$ и $\mathbf{f} = (f_1, f_2, \dots, f_n)$, систему (4.18) запишем в векторной форме

$$\mathbf{f}(\mathbf{x}) = 0. \quad (4.19)$$

Решение системы (4.19) является значительно более сложной задачей, чем решение одного уравнения. Такие системы решают практически только итерационными методами. Обобщим для случая системы (4.19) метод последовательных приближений и метод Ньютона—Рафсона.

Метод простой итерации. Заменим нелинейную систему (4.19) эквивалентной системой специального вида

$$\mathbf{x} = \varphi(\mathbf{x}), \quad (4.20)$$

где $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_n)$.

Предположим, что система (4.20) имеет в ограниченной выпуклой замкнутой области D n -мерного пространства X единственное решение $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$, а компоненты x_i^0 вектора $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$ суть числа, соответственно близкие к x_i^* ($i = 1, 2, \dots, n$).

Будем находить последующие приближения к точному решению с помощью метода простой итерации (последовательных приближений) по формулам:

$$\mathbf{x}^{k+1} = \varphi(\mathbf{x}^k) \quad (4.21)$$

или в координатной форме

$$x_i^{k+1} = \varphi_i(x_1^k, x_2^k, \dots, x_n^k) \quad (i = 1, 2, \dots, n), \quad (k = 0, 1, 2, \dots).$$

Пусть $\rho(\mathbf{x}^1, \mathbf{x}^2) = \|\mathbf{x}^1 - \mathbf{x}^2\|$ — расстояние между элементами \mathbf{x}^1 и \mathbf{x}^2 в пространстве X , где в качестве нормы вектора можно выбрать любую каноническую норму (см. п. 3.3).

По принципу сжатых отображений (см. п. 3.6) система уравнений (4.20) имеет единственное решение $\mathbf{x}^* \in D$, которое может быть найдено методом итераций (4.21) при любом выборе начального приближения $\mathbf{x}^0 \in D$, если все последовательные приближения $\mathbf{x}^k \in D$ ($k = 1, 2, \dots$) и отображение $\varphi(\mathbf{x})$ является сжимающим в D .

Укажем достаточные признаки сходимости метода итераций, удобные в практических вычислениях.

Предположим, что в некоторой выпуклой замкнутой области D функции $\varphi_i(\mathbf{x})$ имеют непрерывные частные производные $\frac{\partial \varphi_i}{\partial x_j}$ и в области D система (4.20) имеет единственное решение \mathbf{x}^* . Пусть для любого начального приближения $\mathbf{x}^0 \in D$ все последующие приближения $\mathbf{x}^k \in D$.

В окрестности решения \mathbf{x}^* по обобщенной формуле Лагранжа

$$x_i^{k+1} - x_i^* = \varphi_i(\mathbf{x}^k) - \varphi_i(\mathbf{x}^*) = \sum_{j=1}^n \frac{\partial \varphi_i}{\partial x_j}(p_i^k)(x_j^k - x_j^*) \quad (i = 1, 2, \dots, n). \quad (4.22)$$

Поэтому

$$\begin{aligned} \rho(\mathbf{x}^{k+1}, \mathbf{x}^*) &= \rho(\varphi(\mathbf{x}^k), \varphi(\mathbf{x}^*)) = \|\varphi(\mathbf{x}^k) - \varphi(\mathbf{x}^*)\| = \\ &= \|J(P^k) \cdot (\mathbf{x}^k - \mathbf{x}^*)\| \leq \|J(P^k)\| \cdot \rho(\mathbf{x}^k, \mathbf{x}^*), \end{aligned} \quad (4.23)$$

где $J(x)$ — матрица Якоби системы (4.20)

$$J(x) = \begin{pmatrix} \frac{\partial \varphi_1(x)}{\partial x_1} & \frac{\partial \varphi_1(x)}{\partial x_2} & \dots & \frac{\partial \varphi_1(x)}{\partial x_n} \\ \frac{\partial \varphi_2(x)}{\partial x_1} & \frac{\partial \varphi_2(x)}{\partial x_2} & \dots & \frac{\partial \varphi_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \varphi_n(x)}{\partial x_1} & \frac{\partial \varphi_n(x)}{\partial x_2} & \dots & \frac{\partial \varphi_n(x)}{\partial x_n} \end{pmatrix} \quad (4.24)$$

Согласно принципу сжатых отображений (см. п. 3.6) метод последовательных приближений (4.21) сходится к решению \mathbf{x}^* системы (4.20), если любая согласованная норма матрицы Якоби $\|J(P^k)\|$ будет меньше единицы.

На практике удобнее рассматривать матрицу M с элементами

$$M_{ij} = \max_D \left| \frac{\partial \varphi_i}{\partial x_j} \right|,$$

норма которой мажорирует норму $\|J(x)\|$.

Запишем неравенство (4.23) в виде

$$\rho(\varphi(\mathbf{x}^k), \varphi(\mathbf{x}^*)) \leq \|J(P^k)\| \cdot \rho(\mathbf{x}^k, \mathbf{x}^*) \leq \|M\| \cdot \rho(\mathbf{x}^k, \mathbf{x}^*). \quad (4.25)$$

Отображение (4.20) будет сжимающим в D , если для любой согласованной нормы матрицы M выполняется условие:

$$\|M\| < 1. \quad (4.26)$$

Близость вектора \mathbf{x}^k к решению \mathbf{x}^* будем характеризовать одной из норм $\|\mathbf{x}\|_\infty$, $\|\mathbf{x}\|_1$ или $\|\mathbf{x}\|_2$.

Учитывая (4.26) и (3.37), для сходимости метода итераций (4.21) достаточно выполнения одного из условий:

$$\begin{aligned} \max_{1 \leq i \leq n} \sum_{j=1}^m M_{ij} &= q_i < 1; \\ \max_{1 \leq j \leq n} \sum_{i=1}^n M_{ij} &= q_j < 1; \\ \left(\sum_{i,j=1}^n M_{ij}^2 \right)^{1/2} &= q < 1. \end{aligned} \quad (4.27)$$

Если $\|M\| \leq \frac{1}{2}$, то для определения решения системы (4.20) с точностью ε итерационный процесс (4.21) продолжаем до выполнения условия

$$\|\mathbf{x}^k - \mathbf{x}^{k-1}\| \leq \varepsilon.$$

Если $\frac{1}{2} < \|M\| < 1$, то решение будет вычислено с точностью ϵ , если для каждой координаты будут выполнены неравенства (4.7).

4.4. Лабораторная работа 3. Приближенное решение системы двух нелинейных уравнений методом простой итерации

Пусть для системы двух уравнений

$$\begin{cases} F_1(x, y) = 0, \\ F_2(x, y) = 0 \end{cases} \quad (4.28)$$

требуется найти действительные решения с заданной точностью ϵ .

Если на плоскости нетрудно построить графики функций $F_1(x, y) = 0$ и $F_2(x, y) = 0$, то можно оценить число корней системы (4.28) и их приближенные значения как координаты точек пересечения графиков.

Преобразуем систему (4.28) к виду, удобному для применения метода простой итерации:

$$\begin{cases} x = \varphi_1(x, y), \\ y = \varphi_2(x, y). \end{cases} \quad (4.29)$$

В методе простой итерации воспользуемся модификацией Зейделя, т. е. построим итерационный процесс по формулам:

$$\begin{cases} x^{k+1} = \varphi_1(x^k, y^k), \\ y^{k+1} = \varphi_2(x^{k+1}, y^k) \quad (k = 0, 1, \dots). \end{cases} \quad (4.30)$$

Пусть в области $D = \{(x; y) | a < x < b; c < y < d\}$ содержится единственное решение $x = x^*$, $y = y^*$ системы (4.29). Тогда достаточные условия сходимости метода простой итерации (4.27) для такого частного случая переформулируются следующим образом.

Если:

- 1) функции $\varphi_1(x, y)$ и $\varphi_2(x, y)$ определены и непрерывно дифференцируемы в D ;
- 2) начальное приближение $(x^0; y^0)$ и все последующие приближения $(x^k; y^k)$ ($k = 1, 2, \dots$) принадлежат D ;
- 3) в D выполнены неравенства

$$\begin{aligned} \left| \frac{\partial \varphi_1}{\partial x} \right| + \left| \frac{\partial \varphi_2}{\partial x} \right| &\leq q_1 < 1, \\ \left| \frac{\partial \varphi_1}{\partial y} \right| + \left| \frac{\partial \varphi_2}{\partial y} \right| &\leq q_2 < 1, \end{aligned} \quad (4.31)$$

или

$$\begin{aligned} \left| \frac{\partial \varphi_1}{\partial x} \right| + \left| \frac{\partial \varphi_1}{\partial y} \right| &\leq q_1 < 1, \\ \left| \frac{\partial \varphi_2}{\partial x} \right| + \left| \frac{\partial \varphi_2}{\partial y} \right| &\leq q_2 < 1, \end{aligned} \quad (4.32)$$

или

$$\sqrt{\sum_{i=1}^2 \left(\left(\frac{\partial \varphi_i}{\partial x} \right)^2 + \left(\frac{\partial \varphi_i}{\partial y} \right)^2 \right)} \leq q < 1, \quad (4.33)$$

то итерационный процесс (4.30) сходится к решению $(x^*; y^*)$ системы.

Погрешность k -го приближения

$$\rho(x^k, x^*) \leq \frac{\alpha}{1-\alpha} \rho(x^k, x^{k-1}), \quad (4.34)$$

где $\alpha = \max\{q_1, q_2\}$ в (4.31), (4.32) или $\alpha = q$ в (4.33).

Если в качестве расстояния выбрать евклидову норму, то оценку (4.34) можно записать в виде

$$\sqrt{(x^k - x^*)^2 + (y^k - y^*)^2} \leq \frac{\alpha}{1-\alpha} \sqrt{(x^k - x^{k-1})^2 + (y^k - y^{k-1})^2}. \quad (4.35)$$

Замечание. Выбор итерирующих функций $\varphi_1(x, y)$ и $\varphi_2(x, y)$ в системе (4.29), обеспечивающий выполнение достаточных условий сходимости итерационного процесса (4.31)–(4.33), является трудоемким процессом. Этот выбор можно осуществить с помощью приема, предложенного в [22]. Полагаем:

$$\begin{aligned} \varphi_1(x, y) &= x + \alpha F_1(x, y) + \beta F_2(x, y); \\ \varphi_2(x, y) &= y + \gamma F_1(x, y) + \delta F_2(x, y), \quad \alpha\delta \neq \beta\gamma. \end{aligned}$$

Коэффициенты α, β, γ и δ определяем как приближение решения системы уравнений:

$$\begin{aligned} 1 + \alpha \frac{\partial F_1(x^0, y^0)}{\partial x} + \beta \frac{\partial F_2(x^0, y^0)}{\partial x} &= 0; \\ \alpha \frac{\partial F_1(x^0, y^0)}{\partial y} + \beta \frac{\partial F_2(x^0, y^0)}{\partial y} &= 0; \\ \gamma \frac{\partial F_1(x^0, y^0)}{\partial x} + \delta \frac{\partial F_2(x^0, y^0)}{\partial x} &= 0; \\ 1 + \gamma \frac{\partial F_1(x^0, y^0)}{\partial y} + \delta \frac{\partial F_2(x^0, y^0)}{\partial y} &= 0. \end{aligned}$$

Если частные производные функций $F_1(x, y)$ и $F_2(x, y)$ не сильно изменяются в окрестности точки $(x^0; y^0)$, то при таком выборе итерирующих функций условия (4.31) — (4.33) выполняются.

Программа 9 решения системы (4.29) двух уравнений методом простой итерации с модификацией Зейделя по формулам (4.30):

Адресс	Команда	Код	Содержание операции
00	ПП	53	
01	17	17	
02	ИП1	61	
03	ИПА	6—	
04	П1	41	
05	—	11	Вычисление погрешности двух приближений
06	FX ²	22	
07	ИП2	62	
08	ИПВ	6L	
09	П2	42	
10	—	11	
11	FX ²	22	
12	+	10	
13	F V ⁻	21	
14	С/П	50	Индикация значения ρ .
15	БП	51	Возврат к началу программы.
16	00	00	
17	Подпрограмма для вычисления
...	$\varphi_1(x^k, y^k), \varphi_2(x^{k+1}, y^k)$.
...	В/0	52	

Распределение регистров памяти:

Номер регистра	1	2	A	B
Переменная	x^k	y^k	x^{k+1}	y^{k+1}

Инструкция. 1. Составить подпрограмму для вычисления $\varphi_1(x, y)$ и $\varphi_2(x, y)$. При этом аргументы x и y функции

$\varphi_1(x, y)$ вызываются из Р1 и Р2, значение функции засыпается в РА. Аргументы x и y функции $\varphi_2(x, y)$ вызываются из РА и Р2, значение функции засыпается в РВ.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В автоматическом режиме (F АВТ) ввести начальное приближение: (x^0) П1 (y^0) П2.

4. Нажать клавиши В/О С/П. После останова на индикаторе высвечивается значение ρ для первого шага. Нажатие клавиши С/П приведет к выполнению следующего шага итерационного процесса.

Продолжаем вычисления до совпадения двух последовательных приближений с заданной точностью ϵ .

5. Значения x^{k+1} и y^{k+1} вызываем из РА и РВ соответственно.

Контрольный пример. Решить систему

$$\begin{cases} x = \frac{1}{2} \ln(x^2 + y^2) + 2,1, \\ y = \operatorname{arctg} \frac{y}{x} + 0,1, \end{cases}$$

выбрав в качестве начального приближения точку $M_0(3; 0)$. Вычисления проводить до совпадения двух последовательных приближений с точностью $\epsilon = 10^{-4}$.

Подпрограмма вычисления значений функций:

ИП1	FX ²	ИП2	FX ²	+	F ln	2	÷	2	F,
1	+	ПА	ИП2	ИПА	÷	F arctg	0	F,	1
+	ПВ	B/O							

Вводим исходные данные 3 П1 0 П2 и осуществляем пуск программы: В/О С/П. Результаты выполнения каждой итерации удобно записывать в таблицу:

Номер шага	0	1	2	3	4	5	6	7
ρ_k	—	0,22237	0,07149	0,02226	0,00686	0,00210	0,00064	0,00020
x^k	3	3,19861	3,26320	3,28351	3,28898	3,29177	3,29237	3,29256
y^k	0	0,1000	0,13063	0,13976	0,14246	0,14325	0,14348	0,14355

На 8-м шаге получим: $\rho_8 = 0,00006$, $x^8 = 3,29261$, $y^8 = 0,14357$.

В промежуточных результатах сохраняем одну запасную цифру. Окончательный ответ округляем до 4-х знаков после запятой.

Ответ: $x = 3,2926$, $y = 0,1436$.

Время вычислений на одном шаге около 15 с.

Задания к лабораторной работе 3. Методом простой итерации с модификацией Зейделя (4.30) решить системы, выбрав в качестве начального приближения точку M_0 . Итерации проводить до совпадения двух последовательных приближений с точностью $\epsilon = 10^{-4}$.

- 1) $\begin{cases} x = e^x \cos y - 1, \\ y = e^x \sin y + 1, \end{cases} M_0(-0,9; 1,4);$
- 2) $\begin{cases} x = \frac{(x^2 - y^2)^2}{4} - x^2 y^2 + 0,5, \\ y = xy(x^2 - y^2) + 0,5, \end{cases} M_0(1; 1);$
- 3) $\begin{cases} x = \frac{x}{x^2 + y^2} + 0,4, \\ y = -\frac{y}{x^2 + y^2} + 1,4, \end{cases} M_0(1; 1);$
- 4) $\begin{cases} x = x^2 + 0,8y^2 + 0,1, \\ y = 2xy + 0,1, \end{cases} M_0(0; 0);$
- 5) $\begin{cases} x = x^2 - y^2 + 0,1, \\ y = 2xy + 0,1, \end{cases} M_0(0; 0);$
- 6) $\begin{cases} x = x^2 - y^2 - 0,1, \\ y = 2xy + 0,1, \end{cases} M_0(0; 0);$
- 7) $\begin{cases} x = x^2 + y^2 + 0,1, \\ y = 2xy - 0,1, \end{cases} M_0(0; 0);$
- 8) $\begin{cases} x = 1 - e^{-x} \cos y, \\ y = e^{-x} \sin y + 1, \end{cases} M_0(0,9; 1,4);$
- 9) $\begin{cases} x = x^2 + y^2 - 0,1, \\ y = 2xy - 0,1, \end{cases} M_0(0; 0);$
- 10) $\begin{cases} x = \frac{x}{x^2 + y^2} + 0,4, \\ y = \frac{1-y}{x^2 + y^2} + 1, \end{cases} M_0(1; 1);$
- 11) $\begin{cases} x = x^2 y^2 - \frac{(x^2 - y^2)^3}{4} - 0,5, \\ y = xy(y^2 - x^2) + 0,5, \end{cases} M_0(-0,5; 0,5);$
- 12) $\begin{cases} x = \frac{x}{x^2 + y^2} - 0,4, \\ y = 1,4 - \frac{y}{x^2 + y^2}, \end{cases} M_0(-1; 1);$
- 13) $\begin{cases} x = -x^2 - 0,8y^2 - 0,1, \\ y = 0,1 - 2xy, \end{cases} M_0(0; 0);$
- 14) $\begin{cases} x = y^2 - x^2 - 0,1, \\ y = 0,1 - 2xy, \end{cases} M_0(0; 0);$
- 15) $\begin{cases} x = 0,1 - x^2 + y^2, \\ y = 0,1 - 2xy, \end{cases} M_0(0; 0);$
- 16) $\begin{cases} x = -x^2 - y^2 - 0,1 \\ y = -2xy - 0,1, \end{cases} M_0(0; 0);$
- 17) $\begin{cases} x = 0,1 - x^2 - y^2, \\ y = -2xy - 0,1, \end{cases} M_0(0; 0);$
- 18) $\begin{cases} x = \frac{x}{x^2 + y^2} - 0,4, \\ y = \frac{1-y}{x^2 + y^2} + 1, \end{cases} M_0(-1; 1);$
- 19) $\begin{cases} x = e^x \cos y - 1, \\ y = e^x \sin y - 1, \end{cases} M_0(-0,9; -1,4);$
- 20) $\begin{cases} x = \frac{(x^2 - y^2)^2}{4} - x^2 y^2 + 0,5, \\ y = xy(x^2 - y^2) - 0,5, \end{cases} M_0(0,5; -0,5);$
- 21) $\begin{cases} x = \frac{x}{x^2 + y^2} - 0,4, \\ y = -\frac{y}{x^2 + y^2} - 1,4, \end{cases} M_0(1; -1);$
- 22) $\begin{cases} x = x^2 + 0,8y^2 + 0,1, \\ y = 2xy - 0,1, \end{cases} M_0(0; 0);$
- 23) $\begin{cases} x = x^2 - y^2 + 0,1, \\ y = 2xy - 0,1, \end{cases} M_0(0; 0);$
- 24) $\begin{cases} x = x^2 - y^2 - 0,1, \\ y = 2xy - 0,1, \end{cases} M_0(0; 0);$
- 25) $\begin{cases} x = x^2 + y^2 + 0,1, \\ y = 2xy + 0,1, \end{cases} M_0(0; 0);$
- 26) $\begin{cases} x = x^2 + y^2 - 0,1, \\ y = 2xy + 0,1, \end{cases} M_0(0; 0);$
- 27) $\begin{cases} x = \frac{x}{x^2 + y^2} + 0,4, \\ y = -\frac{1+y}{x^2 + y^2} - 1, \end{cases} M_0(1; -1);$
- 28) $\begin{cases} x = \cos xy, \\ y = x^2 - 1, \end{cases} M_0(1; 1);$

$$29) \begin{cases} x = \frac{1}{2} \ln(x^2 + y^2) + 2,1, \\ y = \frac{y}{x} - 0,1, \quad M_0(3; 0); \end{cases} \quad 30) \begin{cases} x = y^2 - x^2 - 0,1, \\ y = -2xy - 0,1, \quad M_0(0; 0). \end{cases}$$

4.5. Решение систем нелинейных уравнений методом Ньютона

Если в системе нелинейных уравнений $\mathbf{f}(\mathbf{x}) = 0$ существуют частные производные функций f_i по переменным x_i , то для ее решения применим метод Ньютона. Через

$$J(\mathbf{x}) = \left[\frac{\partial f_i}{\partial x_j} \right] \quad (i, j = 1, 2, \dots, n)$$

обозначим матрицу Якоби системы (4.19). Выберем нулевое приближение \mathbf{x}^0 в достаточно близкой окрестности корня \mathbf{x}^* . Линеаризуем вектор-функцию $\mathbf{f}(\mathbf{x})$, разлагая ее в ряд Тейлора и ограничиваясь лишь членами нулевой и первой степени. Тогда на k -м шаге $(k+1)$ -е приближение будет решением уравнения

$$\mathbf{f}(\mathbf{x}^k) + J(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k) = 0 \quad (k = 0, 1, 2, \dots). \quad (4.36)$$

В частности, если существует матрица $J^{-1}(\mathbf{x}^k)$, обратная матрице Якоби, то итерационный процесс Ньютона (4.36) можно записать в виде:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - J^{-1}(\mathbf{x}^k)\mathbf{f}(\mathbf{x}^k) \quad (k = 0, 1, 2, \dots). \quad (4.37)$$

Формула (4.37) обобщает формулу Ньютона (4.8) для одного уравнения.

Если вторые частные производные функции $\mathbf{f}(\mathbf{x})$ непрерывны вблизи \mathbf{x}^* , $\det J(\mathbf{x}) \neq 0$ в окрестности корня, а начальное приближение \mathbf{x}^0 близко к решению, то метод Ньютона сходится, причем сходимость будет квадратичной [19] (более подробно о выборе начального приближения и об условиях сходимости метода Ньютона [14]).

Ввиду трудоемкости вычисления и обращения матрицы Якоби на каждом шаге можно вычислить на первом шаге $J^{-1}(\mathbf{x}^0)$ и использовать модифицированный метод Ньютона:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - J^{-1}(\mathbf{x}^0)\mathbf{f}(\mathbf{x}^k) \quad (k = 0, 1, 2, \dots). \quad (4.38)$$

Сходимость такого процесса будет уже линейной [19].

Замечание. При решении систем нелинейных уравнений иногда строят функционал, минимум которого достигается на решении системы. Например, нулевой минимум функции

$F(\mathbf{x}) = \sum_{i=1}^n |f_i(\mathbf{x})|^2$, где $\mathbf{x} = (x_1, x_2, \dots, x_n)$, является решением системы $\mathbf{f}(\mathbf{x}) = 0$. Минимум функционала приближенно определяют каким-либо из методов спуска (см. гл. 6), которые обычно имеют более широкую область сходимости. Полученное приближение уточняют затем методом Ньютона.

4.6. Лабораторная работа 4. Приближенное решение системы двух нелинейных уравнений методом Ньютона

Рассмотрим систему двух уравнений

$$\begin{cases} u(x, y) = 0, \\ v(x, y) = 0, \end{cases} \quad (4.39)$$

где функции $u(x, y)$ и $v(x, y)$ — непрерывно дифференцируемы в области D , содержащей единственное решение $x = x^*$, $y = y^*$. Выберем в области D начальное приближение $(x^0; y^0)$ и будем вычислять последовательные приближения согласно методу Ньютона (4.37):

$$\begin{cases} x^{k+1} = x^k - \frac{\Delta_x^k}{\Delta_y^k}, \\ y^{k+1} = y^k - \frac{\Delta_y^k}{\Delta_x^k}. \end{cases} \quad (4.40)$$

$(k = 0, 1, 2, \dots),$

где

$$\Delta^k = \det J(x^k, y^k) = \begin{vmatrix} u'_x(x^k, y^k) & u'_y(x^k, y^k) \\ v'_x(x^k, y^k) & v'_y(x^k, y^k) \end{vmatrix} \text{ — якобиан,}$$

$$\Delta_x^k = \begin{vmatrix} u(x^k, y^k) & u'_y(x^k, y^k) \\ v(x^k, y^k) & v'_y(x^k, y^k) \end{vmatrix}, \quad \Delta_y^k = \begin{vmatrix} u'_x(x^k, y^k) & u(x^k, y^k) \\ v'_x(x^k, y^k) & v(x^k, y^k) \end{vmatrix}.$$

В качестве критерия окончания итерационного процесса (4.40) выберем условие

$$\rho_k = \frac{1}{|\Delta^k|} \sqrt{(\Delta_x^k)^2 + (\Delta_y^k)^2} < \varepsilon, \quad (4.41)$$

где ρ_k — евклидово расстояние между двумя последовательными приближениями.

Программа 10 решения системы (4.39) двух нелинейных уравнений методом Ньютона по формулам (4.40):

Адрес	Команда	Код	Содержание операции
00	ПП	53	
01	41	41	Вызов подпрограммы вычисления функций $a(x, y)$, $v(x, y)$ и элементов матрицы Якоби
02	ИП2	62	}
03	ИП7	67	}
04	×	12	
05	ИП3	63	
06	ИП6	66	Вычисление Δ^k
07	×	12	
08	—	11	
09	П4	44	
10	8	08	
11	ПО	40	Организация цикла
12	КИПО	Г0	}
13	ИП8	68	
14	×	12	
15	КИПО	Г0	
16	ИП9	69	
17	×	12	Вычисление $\frac{\Delta_x^k}{\Delta^k}$ и $-\frac{\Delta_y^k}{\Delta^k}$.
18	—	11	
19	ИП4	64	
20	÷	13	
21	КП0	Л0	
22	FЛ0	5Г	
23	12	12	Проверка окончания цикла
24	ИПА	6—	
25	ИГ15	65	
26	—	11	Вычисление x^{k+1}
27	ПА	4—	
28	ИПВ	6Л	
29	ИП1	61	
30	+	10	Вычисление y^{k+1}
31	ПВ	4Л	

Адресс	Команда	Ход	Содержание операции
32	ИП5	65	
33	FX ²	22	
34	ИП1	61	
35	FX ²	22	Вычисление ρ_k
36	+	10	
37	F V ⁻	21	
38	С/П	50	Индикация ρ_k
39	БП	51	
40	00	00	Возврат к началу программы
41	
...	
...	B/O	52	Подпрограмма для вычисления $u, v, u'_x, u'_y, v'_x, v'_y$

Распределение регистров памяти:

Номер регистра	0	1	2	3	4	5	6	7	8	9	A	B
Перемен-ная	счетчик цикла	$-\frac{\Delta_y^k}{\Delta^k}$	u'_x	v'_x	Δ^k	$\frac{\Delta_x^k}{\Delta^k}$	u'_y	v'_y	u	v	x^{k+1}	y^{k+1}

Свободны РС и РД.

Инструкция. 1. Составить подпрограмму вычисления функций $u, v, u'_x, u'_y, v'_x, v'_y$, значения x и y вызываются из РА и РВ соответственно, результаты вычислений засыпаются соответственно в регистры Р8, Р9, Р2, Р6, Р3, Р7.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В автоматическом режиме (F АВТ) ввести начальное приближение (x^0) ПА (y^0) ПВ.

4. Перейти к вычислениям, нажав клавиши В/О С/П. После ОСТАНОВа на индикаторе высвечивается значение ρ_0 . Нажатие клавиши С/П обеспечивает выполнение следующего шага метода Ньютона. Вычисления продолжаем до совпадения двух последовательных приближений с точностью ϵ , т. е. $\rho_k \leq \epsilon$.

5. Значения x и y вызываем из РА и РВ соответственно, значения функций $u(x, y)$ и $v(x, y)$ вызываем из Р8 и Р9.

Контрольный пример. Методом Ньютона решить систему

$$\begin{cases} e^x - e^y + 1 = 0, \\ xy - 1 = 0, \end{cases}$$

выбрав в качестве начального приближения $M_0(2; 2)$. Итерации продолжать до совпадения двух последовательных приближений с точностью $\epsilon = 10^{-4}$.

Составим подпрограмму для вычисления значений функций $u(x, y) = e^x - e^y + 1$, $v(x, y) = xy - 1$ и их частных производных $u'_x = e^x$, $u'_y = -e^y$, $v'_x = y$, $v'_y = x$:

ИПА	П7	Fe^X	П2	ИПВ	П3	Fe^X	/-/	П6	+
I	+	П8	ИПА	ИПВ	×	1	—	Р9	B/O

Вводим исходные данные 2 ПА ПВ и осуществляя пуск программы В/О С/П. Результаты выполнения каждого шага метода Ньютона запишем в таблицу:

Номер шага, k	0	1	2	3	4	5
p_k	—	1,06497	0,33398	0,03829	0,00043	$2,48 \cdot 10^{-8}$
x^k	2	1,18233	0,37393	0,83614	0,83604	0,83604
y^k	2	1,31267	1,18949	1,19569	1,19611	1,19611

На последнем шаге итерации вызываем из Р8 и Р9 значения функций:

$$u(x^k, y^k) = -1 \cdot 10^{-7}, v(x^k, y^k) = 0.$$

Ответ: $x = 0,8360$; $y = 1,1961$.

Время вычислений на одном шаге около 23 с.

Задания к лабораторной работе 4. Решить системы уравнений методом Ньютона, выбрав в качестве начального приближения точку $M_0(x^0; y^0)$. Итерации продолжать до совпадения двух последовательных приближений с точностью $\epsilon = 10^{-4}$.

$$1) \begin{cases} e^y - e^x + 1 = 0, \\ xy - 1 = 0, \quad M_0(2; 2); \end{cases}$$

$$7) \begin{cases} e^x + e^y - 1 = 0, \\ x^3 - y^3 + 1 = 0, \\ M_0(-1; -0,5); \end{cases}$$

$$2) \begin{cases} e^x - y^2 + 1 = 0, \\ xy - 1 = 0, \quad M_0(2; 2); \end{cases}$$

$$8) \begin{cases} \sin x + \sin y - 1 = 0, \\ x^2 - y^2 - 1 = 0, \quad M_0(1; 0); \end{cases}$$

$$3) \begin{cases} y + \lg x = 0, \quad M_0(0; 1); \\ xy^2 - 1 = 0, \quad M_0(0; 1); \end{cases}$$

$$9) \begin{cases} e^x - e^y + 1 = 0, \\ xy - 1 = 0, \quad M_0(2; 2); \end{cases}$$

$$4) \begin{cases} y - e^x = 0, \quad M_0(0,5; 1,5); \\ xy^2 - 1 = 0, \quad M_0(0,5; 1,5); \end{cases}$$

$$10) \begin{cases} e^y - x^2 + 1 = 0, \\ xy - 1 = 0, \quad M_0(2; 2); \end{cases}$$

$$5) \begin{cases} e^x - e^y + 1 = 0, \\ x^3 + y^3 - 1 = 0, \quad M_0(0,5; 1); \end{cases}$$

$$11) \begin{cases} e^y - x^2 + 1 = 0, \\ x + \lg y = 0, \quad M_0(1; 0); \end{cases}$$

$$6) \begin{cases} e^x - e^y + 1 = 0, \\ x^3 + y^3 + 1 = 0, \quad M_0(-1; 1); \end{cases}$$

- 12) $\begin{cases} x^2y - 1 = 0, \\ x - e^y = 0, \end{cases} M_0(1,5; 0,5);$ 20) $\begin{cases} xy^2 + 1 = 0, \\ y - e^{-x} = 0, \end{cases} M_0(-0,5; 1,5);$
 13) $\begin{cases} e^{-x} - e^y - 1 = 0, \\ x^3 + y^3 - 1 = 0, \end{cases} M_0(1; 0,5);$ 21) $\begin{cases} e^{-x} - e^y + 1 = 0, \\ x^3 - y^3 + 1 = 0, \end{cases} M_0(-0,5; 1);$
 14) $\begin{cases} x^3 + y^3 + 1 = 0, \\ e^x + e^y - 1 = 0, \end{cases} M_0(1; -1);$ 22) $\begin{cases} e^{-x} - e^y + 1 = 0, \\ x^3 - y^3 - 1 = 0, \end{cases} M_0(1; 1);$
 15) $\begin{cases} x^3 - y^3 - 1 = 0, \\ \sin x + \sin y - 1 = 0, \end{cases} M_0(-0,5; -1);$ 23) $\begin{cases} e^{-x} + e^y - 1 = 0, \\ x^3 + y^3 - 1 = 0, \end{cases} M_0(1; -0,5);$
 16) $\begin{cases} x^2 - y^2 + y = 0, \\ \sin x - \sin y + 1 = 0, \end{cases} M_0(0; 1);$ 24) $\begin{cases} x^2 - y^2 - 1 = 0, \\ \sin x - \sin y + 1 = 0, \end{cases} M_0(-1; 0);$
 17) $\begin{cases} xy + 1 = 0, \\ e^{-x} - y^2 + 1 = 0, \end{cases} M_0(-2; 2);$ 25) $\begin{cases} e^{-y} - e^x + 1 = 0, \\ xy + 1 = 0, \end{cases} M_0(2; -2);$
 18) $\begin{cases} xy + 1 = 0, \\ e^{-x} - y^2 - 1 = 0, \end{cases} M_0(-2; 2);$ 26) $\begin{cases} xy^2 - 1 = 0, \\ y - \operatorname{tg} x = 0, \end{cases} M_0(0,5; -1,5).$
 19) $\begin{cases} y - \operatorname{tg} x = 0, \end{cases} M_0(0; 1);$

Глава 5. ПРИБЛИЖЕНИЕ ФУНКЦИЙ

5.1. Постановка задачи приближения функций

Задача приближения (аппроксимации) заключается в замене функциональной зависимости, заданной на множестве $X \subseteq \mathbb{R}$ в виде таблицы, графика, формулы или в неявном виде, более простой приближающей функцией.

Если X состоит из дискретного множества точек x_0, x_1, \dots, x_m , то приближение называется *точечным*, а если это отрезок $[a; b]$, то приближение называется *интегральным*.

Пусть функция $f(x)$ аппроксимируется на множестве X функцией $\varphi(x)$, более удобной для вычислений и близкой в некотором смысле к $f(x)$. Качество аппроксимации, т. е. близость двух функций, можно оценивать по-разному в зависимости от физического содержания решаемой задачи. В соответствии с этим в теории приближений рассматриваются различные функциональные пространства с введенной в них метрикой, позволяющей оценить расстояние между функциями $\rho(f, \varphi)$.

Пространство L называется *нормированным*, если каждому элементу $f \in L$ поставлено в соответствие действительное число $\|f\|$, называемое *нормой*, удовлетворяющее следующим аксиомам:

1) $\|f\| \geq 0$, $\|f\| = 0$ тогда и только тогда, если $f = 0$;

2) $\|\alpha \cdot f\| = |\alpha| \|f\|$, где $\alpha \in \mathbb{R}$;

3) $\|f + \varphi\| \leq \|f\| + \|\varphi\|$.

Полное линейное нормированное пространство называется *банаховым* или *B-пространством*.

Нормированное пространство становится метрическим, если положить $\rho(f, \varphi) = \|f - \varphi\|$.

Вещественное линейное пространство L называется *евклидовым*, если каждой паре его элементов f и φ поставлено в соответствие вещественное число (f, φ) , называемое *скалярным произведением*, удовлетворяющее следующим аксиомам:

- 1) $(f, f) \geq 0$, $(f, f) = 0$ тогда и только тогда, если $f = 0$;
- 2) $(f, \varphi) = (\varphi, f)$;
- 3) $(\alpha f, \varphi) = \alpha (f, \varphi)$, $\alpha \in \mathbb{R}$;
- 4) $(f + \varphi, g) = (f, g) + (\varphi, g)$.

Полное линейное евклидово пространство L называется *гильбертовым*. Гильбертово пространство является банаховым. Норму функции $f(x) \in L$ можно определить через скалярное произведение

$$\|f\| = \sqrt{(f, f)}.$$

Конечная система элементов $\varphi_0, \varphi_1, \dots, \varphi_n$ пространства L называется *линейно независимой*, если их линейная комбинация является нулевым элементом пространства L , т. е. выполняется равенство

$$c_0\varphi_0 + c_1\varphi_1 + \dots + c_n\varphi_n = 0$$

лишь при условии $c_0 = c_1 = \dots = c_n = 0$. В противном случае эта система называется *линейно зависимой*.

Бесконечная система элементов $\varphi_0, \varphi_1, \varphi_2, \dots$ называется *линейно независимой*, если любая ее подсистема, состоящая из конечного числа элементов, линейно независима.

Необходимым и достаточным условием линейной независимости конечной системы элементов $\varphi_0, \varphi_1, \dots, \varphi_n$ является отличие от нуля определителя Грамма, т. е.

$$\begin{vmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \dots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \dots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \dots & (\varphi_n, \varphi_n) \end{vmatrix} \neq 0.$$

Если в пространстве L существует система n линейно независимых элементов, а любые $n+1$ элементов этого пространства линейно зависимы, то число n называют *размерностью* пространства L и обозначают $n = \dim L$. Если в L существует счетное или несчетное бесконечное число линейно

независимых элементов, то L называют *бесконечномерным* и полагают $\dim L = \infty$.

Система линейно независимых элементов $\varphi_1, \varphi_2, \dots, \varphi_n$ n -мерного пространства L называется *базисом* этого пространства, если для каждого элемента f пространства L найдутся вещественные числа c_1, c_2, \dots, c_n такие, что справедливо равенство

$$f = \sum_{i=1}^n c_i \varphi_i.$$

Последовательность $\{\varphi_n(x)\}$ функций бесконечномерного банахова пространства L называется *базисом* в L , если каждая функция $f(x) \in L$ раскладывается в ряд

$$f(x) = \sum_{i=0}^{\infty} c_i \varphi_i(x),$$

сходящийся по норме пространства L [20].

В гильбертовом пространстве L вводится понятие ортогональности двух функций. Функции φ_1 и φ_2 называются *ортогональными*, если их скалярное произведение равно нулю, т. е. $(\varphi_1, \varphi_2) = 0$.

Если $\|\varphi\| = 1$, то функцию $\varphi(x)$ называют *нормированной*. Конечная или бесконечная система функций $\varphi_0, \varphi_1, \dots$ называется *ортонормированной*, если ее элементы попарно ортогональны и нормированы, т. е.

$$(\varphi_i, \varphi_k) = \begin{cases} 0, & i \neq k, \\ 1, & i = k. \end{cases}$$

Ортонормированная система функций в гильбертовом пространстве L называется *полной*, если ее нельзя дополнить элементами из L так, чтобы новая система была ортонормированной, т. е. в L не существует ненулевого элемента, ортогонального ко всем элементам данной системы. Полная ортонормированная система функций образует базис в пространстве L .

Если рассматривать линейное пространство вещественных функций, заданных на множестве X , то система его линейно независимых элементов называется *системой линейно независимых на X функций*.

Пусть $f(x)$ является элементом линейного нормированного пространства L . Выберем в L систему $n + 1$ линейно независимых функций $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$ и образуем линейную комбинацию

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$$

с постоянными коэффициентами c_i ($i = 0, 1, \dots, n$). Функция $\varphi(x)$ называется *обобщенным многочленом порядка n* . В частности, если $\varphi_i(x) = x^i$ ($i = 0, 1, \dots, n$), то

$$\varphi(x) = c_0 + c_1x + \dots + c_nx^n$$

является алгебраическим многочленом степени n .

Если выбрать $\varphi_0(x) = 1$, $\varphi_1(x) = \cos x$, $\varphi_2(x) = \sin x$, ..., $\varphi_{2n-1}(x) = \cos nx$, $\varphi_{2n}(x) = \sin nx$, то функция $\varphi(x) = \frac{a_0}{2} + a_1 \cos x + b_1 \sin x + \dots + a_n \cos nx + b_n \sin nx$ называется *тригонометрическим многочленом порядка n* .

Будем аппроксимировать функцию $f(x)$ обобщенным многочленом n -го порядка $\varphi(x)$. Параметры c_i ($i = 0, 1, \dots, n$) обобщенного аппроксимирующего многочлена

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$$

определяются из условия минимума расстояния $\rho(f, \varphi) = \|f - \varphi\|$ в метрике пространства L .

Рассмотрим некоторые функциональные пространства и соответствующие им задачи теории приближений.

1. Пусть функция $f(x)$ принадлежит пространству $C[a; b]$ вещественных функций, непрерывных на отрезке $[a; b]$, в котором *равномерная (чебышевская) норма* вводится следующим образом:

$$\|f\|_{C[a; b]} = \max_{x \in [a; b]} |f(x)|.$$

В соответствии с этой метрикой за меру близости функций $f(x)$ и $\varphi(x)$ принимают их максимальное отклонение во всех отдельно взятых точках отрезка $[a; b]$, т. е.

$$\rho(f, \varphi) \equiv \|f - \varphi\|_{C[a; b]} = \max_{x \in [a; b]} |f(x) - \varphi(x)|. \quad (5.1)$$

Функция $\varphi(x)$, обеспечивающая минимум расстояния ρ , называется *наилучшим равномерным приближением* функции $f(x)$ на отрезке $[a; b]$.

Пространство $C[a; b]$ является полным нормированным, т. е. банаховым пространством.

2. Рассмотрим гильбертово пространство $L_2^p[a; b]$ вещественных функций, интегрируемых с квадратом с весом $p(x)$ на отрезке $[a; b]$, т. е. для $f(x) \in L_2^p[a; b]$ конечен интеграл в смысле Лебега

$$\int_a^b p(x) f^2(x) dx.$$

Предполагается, что на отрезке $[a; b]$ весовая функция $p(x) \geq 0$ и $\int_a^b p(x) dx > 0$. Ее целесообразно вводить в тех случаях, когда функция $f(x)$ аппроксимируется с разной степенью точности в различных точках отрезка $[a; b]$. В частности, если $p(x) = 1$ для всех $x \in [a; b]$, то используют обозначение $L_2[a; b]$.

Скалярное произведение функций в пространстве $L_2^p[a; b]$ и средняя квадратическая норма вводятся следующим образом:

$$(f, \varphi) = \int_a^b p(x) f(x) \varphi(x) dx,$$

$$\|f\|_{L_2^p[a; b]} = (f, f)^{1/2} = \left(\int_a^b p(x) f^2(x) dx \right)^{1/2}.$$

Расстояние между функциями

$$\rho(f, \varphi) = \|f - \varphi\|_{L_2^p[a; b]} = \left(\int_a^b p(x) [f(x) - \varphi(x)]^2 dx \right)^{1/2} \quad (5.2)$$

называется *взвешенным средним квадратическим отклонением* или погрешностью приближения в пространстве $L_2^p[a; b]$. Сходимость функциональной последовательности в смысле метрики пространства $L_2^p[a; b]$ называется *сходимостью в среднем*.

3. Пространство функций, заданных своими значениями на дискретном множестве X точек x_0, x_1, \dots, x_m , принадлежащих отрезку $[a; b]$, будет гильбертовым размерности $n + 1$, если в нем определено скалярное произведение следующим образом:

$$(\varphi_k, \varphi_i) = \sum_{i=0}^m p_i \varphi_k(x_i) \cdot \varphi_i(x_i) \quad (p_i > 0).$$

В этом пространстве вводится норма

$$\|f\|_X = \left(\sum_{i=0}^m p_i f^2(x_i) \right)^{1/2}$$

и взвешенное среднее квадратическое отклонение

$$\rho(f, \varphi) = \|f - \varphi\|_X = \left(\sum_{i=0}^m p_i [f(x_i) - \varphi(x_i)]^2 \right)^{1/2}. \quad (5.3)$$

С помощью положительных весов p_i можно учитывать неодинаковую точность измерения значений функции.

Функция $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$, обеспечивающая минимум квадрата расстояния (5.2) или (5.3), называется соответственно *наилучшим интегральным или точечным взвешенным средним квадратическим приближением* функции $f(x)$.

В интегральном приближении базисные функции $\varphi_i(x)$ ($i = 0, 1, \dots, n$) предполагаются линейно независимыми на $[a; b]$, а в точечном — на множестве точек $\{x_i\}_0^m$, при этом $n \ll m$.

4. Если значения функции $f(x)$ заданы в $n + 1$ точках x_0, x_1, \dots, x_n отрезка $[a; b]$ и достаточно простая для вычислений приближающая ее функция $\varphi(x)$ совпадает с $f(x)$ в этих точках, а в остальных точках отрезка $[a; b]$ приближенно представляет $f(x)$, то такое приближение называется *интерполяцией*.

Ниже изложены методы интерполяции алгебраическими многочленами и сплайнами, а также точечная и интегральная средняя квадратическая аппроксимация функции алгебраическими, ортогональными, в том числе тригонометрическими многочленами по методу наименьших квадратов. Соответствующие вычисления реализуются на микрокалькуляторах.

5.2. Полиномиальная интерполяция

Пусть на отрезке $[x_0; x_n]$ заданы точки x_0, x_1, \dots, x_n (узлы интерполяции) и известны значения функции $y = f(x)$ в этих точках: $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$. Требуется построить функцию $\varphi(x)$, совпадающую в узлах интерполяции со значениями функции $f(x)$.

Таблица значений функции $f(x)$ может быть интерполирована бесконечным множеством различных функций, поэтому нужно иметь некоторый критерий выбора. Многие интерполирующие функции строятся как линейные комбинации элементарных функций. Линейные комбинации одночленов $\{x^k\}_0^n$ приводят к алгебраическим полиномам, линейные комбинации тригонометрических функций $\{\cos kx, \sin kx\}_0^n$ приводят к тригонометрическим полиномам.

Рассмотрим интерполяцию алгебраическими полиномами. В многочисленных приложениях аппарата интерполяции в основном приходится решать две задачи:

1) найти аналитическое выражение для функции, заданной таблицей, графиком или в виде сложной для приложений формулы;

2) вычислить значение функции $f(x)$ в точках, не совпадающих с узлами интерполяции, с помощью сравнительно несложного алгоритма.

Интерполяционные полиномы находят также широкое применение в численном дифференцировании и интегрировании, при решении дифференциальных уравнений.

Интерполяционный полином Лагранжа. Выберем в качестве приближающей функции $\varphi(x)$ полином $P_n(x)$ степени n ,

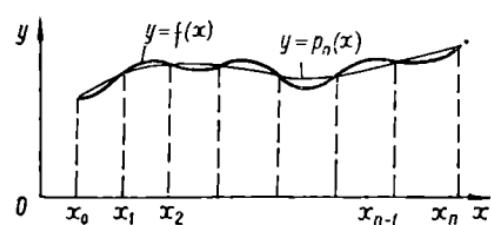


Рис. 11

значения которого совпадают со значениями функции $f(x)$ в узлах интерполяции, т. е.

$$\begin{aligned} P_n(x_0) &= y_0, \quad P_n(x_1) = \\ &= y_1, \dots, \quad P_n(x_n) = y_n. \end{aligned} \quad (5.4)$$

Геометрически это задача о построении параболы n -го порядка $y = P_n(x)$, пересекающейся с графиком функции $y = f(x)$ в $(n+1)$ -й наперед заданной точке (рис. 11). Покажем, что такая задача имеет единственное решение.

Пусть

$$P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n.$$

Коэффициенты a_i ($i = 0, 1, 2, \dots, n$) можно определить из системы уравнений

$$\left\{ \begin{array}{l} a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0, \\ a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1, \\ \dots \dots \dots \dots \dots \dots \dots \\ a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n. \end{array} \right. \quad (5.5)$$

Определитель этой системы

$$\Delta = \begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{vmatrix} = \prod_{1 \leq i < j \leq n} (x_i - x_j)$$

называется *определителем Вандермонда*. Если узлы интерполяции различны, то $\Delta \neq 0$, т. е. система (5.5) имеет единственное решение.

При определении полинома $P_n(x)$ используем базис так

называемых лагранжевых коэффициентов $L_i(x)$ ($i = 0, 1, \dots, n$) степени n таких, что

$$L_i(x_j) = \begin{cases} 1, & \text{если } i = j, \\ 0, & \text{если } i \neq j. \end{cases} \quad (5.6)$$

Определим $L_i(x)$ как многочлен степени n , обращающийся в нуль в точках $x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$ и равный единице в точке x_i , полагая

$$L_i(x) = A(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n).$$

При $x = x_i$

$$1 = A(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n).$$

Определив значение A , получим окончательно

$$L_i(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}. \quad (5.7)$$

Полином $y_i L_i(x)$ принимает значение y_i в точке x_i и равен нулю во всех точках x_j ($j \neq i$).

Обозначим

$$\omega_n(x) = (x - x_0)(x - x_1) \dots (x - x_n).$$

Легко убедиться в том, что

$$\omega'_n(x_i) = (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n).$$

Поэтому выражение (5.7) можно записать в виде

$$L_i(x) = \frac{\omega_n(x)}{(x - x_i)\omega'_n(x_i)}.$$

Лагранжевы коэффициенты $L_i(x)$ называют еще множителями влияния соответствующих узлов интерполяции.

Интерполяционный полином Лагранжа степени n выражается формулой

$$P_n(x) = \sum_{i=0}^n y_i L_i(x) = \sum_{i=0}^n y_i \frac{\omega_n(x)}{(x - x_i)\omega'_n(x_i)}. \quad (5.8)$$

Число арифметических операций для его вычисления пропорционально n^2 . Вычисление значений полинома Лагранжа для числа узлов $n + 1 \leq 9$ осуществляется с помощью программы 11.

Отметим, что интерполяционный полином Лагранжа удобно применять в тех случаях, когда ведется многократное интерполирование по одним и тем же узлам (пусть даже для различных функций). Для этих узлов можно заранее составить коэффициенты Лагранжа $L_i(x)$, поскольку они не зависят от функции $f(x)$.

Однако, если для улучшения приближения потребуется повысить на единицу число узлов прибавлением нового узла, то полином Лагранжа придется пересчитать заново. В этом отношении удобней пользоваться интерполяционным полиномом Ньютона (см. ниже).

Если не требуется находить общее выражение для полинома Лагранжа, а нужно только вычислять его значения для конкретных x , то удобно пользоваться интерполяционной схемой Эйткена [4].

Согласно этой схеме вычислим выражение

$$L_{01}(x) = \frac{\begin{vmatrix} y_0 & x_0 - x \\ y_1 & x_1 - x \end{vmatrix}}{x_1 - x_0},$$

которое является линейным интерполяционным полиномом для узлов x_0 и x_1 . Для другой пары соседних узлов x_1 и x_2 вычислим аналогично

$$L_{12}(x) = \frac{\begin{vmatrix} y_1 & x_1 - x \\ y_2 & x_2 - x \end{vmatrix}}{x_2 - x_1}$$

и т. д.

Выражение

$$L_{012}(x) = \frac{\begin{vmatrix} L_{01}(x) & x_0 - x \\ L_{12}(x) & x_2 - x \end{vmatrix}}{x_2 - x_0}$$

представляет собой квадратичный многочлен, интерполирующий значения y_0, y_1, y_2 соответственно в узлах x_0, x_1, x_2 . Подключая последовательно новые узлы интерполяции, в общем случае получим выражение

$$L_{0123\dots n}(x) = \frac{\begin{vmatrix} L_{012\dots(n-1)}(x) & x_0 - x \\ L_{123\dots n}(x) & x_n - x \end{vmatrix}}{x_n - x_0},$$

которое будет интерполяционным многочленом Лагранжа, принимающим в точках x_0, x_1, \dots, x_n соответственно значения y_0, y_1, \dots, y_n . Вычисление значения этого полинома в точке x по схеме Эйткена удобно проводить, записывая промежуточные результаты в табл. 5.1 (для $n = 4$).

Таблица 5.1

x_i	y_i	$L_{i-1, i}$	$L_{i-2, i-1, i}$	$L_{i-3, i-2, i-1, i}$	$L_{i-4, i-3, i-2, i-1, i}$
x_0	y_0				
x_1	y_1	L_{01}			
x_2	y_2	L_{12}	L_{012}		
x_3	y_3	L_{23}	L_{123}	L_{0123}	
x_4	y_4	L_{34}	L_{234}	L_{1234}	L_{01234}

Описанный рекуррентный процесс интерполяции не требует априорного задания степени интерполяционного полинома. Новые узлы интерполяции подключаются в процессе вычислений, которые обычно проводятся до тех пор, пока последовательные значения $L_{0123 \dots n}(x)$ и $L_{012 \dots n, n+1}(x)$ не совпадут в пределах заданной точности. Вычисления элементов табл. 5.1 по схеме Эйткена реализуются на микрокалькуляторе с помощью программы 12.

Если узлы интерполяции равнодалены друг от друга, т. е. $x_{i+1} - x_i = h = \text{const}$ ($i = 0, 1, 2, \dots, n - 1$), то, введя переменную $t = (x - x_0)/h$, лагранжевы коэффициенты можно записать в виде

$$L_i(t) = (-1)^{n-i} C_n^i \frac{t(t-1)(t-2) \cdots (t-n)}{(t-i)n!}, \quad (5.9)$$

т. е. получить следующее выражение для интерполяционного полинома Лагранжа с равноотстоящими узлами

$$P_n(t) = \sum_{i=0}^n y_i L_i(t) = \frac{t(t-1)(t-2) \cdots (t-n)}{n!} \sum_{i=0}^n (-1)^{n-i} \frac{C_n^i}{t-i} y_i, \quad (5.10)$$

где коэффициенты $L_i(t)$ не зависят от выбора шага h и от функции $f(x)$. Для них составлены специальные таблицы, упрощающие вычисление полинома Лагранжа.

Приведем оценку погрешности интерполяционного полинома Лагранжа [14]. Пусть функция $f(x)$ имеет $n + 1$ непрерывную производную для всех $x \in [x_0; x_n]$. Остаточный член интерполяционного полинома Лагранжа

$$R_n(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_n(x), \quad (5.11)$$

где ξ — внутренняя точка минимального отрезка, содержащего узлы интерполяции x_0, x_1, \dots, x_n и точку x .

Погрешность зависит от выбора узлов x_i , точки x и свойств функции $f(x)$. Известно [14], что величина $\sup | \omega_n(x) |$ будет наименьшей, если в качестве узлов интерполяции $x \in [a; b]$ выбрать корни полиномов Чебышева $T_{n+1}(x) = \cos [(n + \frac{1}{2}) \arccos x]$ для $|x| \leq 1$, определяемые по формулам

$$x_i = \cos \frac{2i + 1}{2n + 2} \pi \quad (i = 0, 1, \dots, n - 1).$$

Но такой выбор узлов редко применяется в практических задачах, поскольку при этом значительно усложняется интерполяция, а выигрыш в точности невелик.

На значение интерполяционного многочлена в точке x наибольшее влияние оказывают ближайшие к точке x узлы интерполяции. Такие узлы обозначим через x_0 и x_1 . Затем постепенно добавляем соседние узлы справа и слева, чтобы они симметрично располагались относительно точки x .

Погрешность интерполяции будет меньшей, если точка x расположена ближе к середине отрезка $[x_0; x_n]$ [19] и может оказаться значительной, если проводить экстраполацию, т. е. выбирать значения x вне отрезка $[x_0; x_n]$.

Интерполяционный полином Ньютона. Укажем форму записи интерполяционного полинома $P_n(x)$, которая допускает уточнение результатов интерполяции последовательным прибавлением новых узлов. При этом будем использовать разделенные и конечные разности функции, являющиеся удобным аппаратом при работе с таблично заданными функциями.

Пусть для функции $f(x)$ заданы не обязательно равноотстоящие узлы интерполяции $x_0, x_1, \dots, x_n, x_i \neq x_j$ при $i \neq j$; $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$ — соответствующие значения функции.

Разделенными разностями первого порядка называются величины, имеющие смысл средних скоростей изменения функции и вычисляемые по формулам:

$$f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \dots, f(x_{n-1}; x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

Аналогично определяются разделенные разности второго порядка

$$f(x_0; x_1; x_2) = \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0},$$

$$f(x_1; x_2; x_3) = \frac{f(x_2; x_3) - f(x_1; x_2)}{x_3 - x_1}.$$

и т. д.

В общем случае разделенные разности k -го порядка определяются через разделенные разности $(k - 1)$ -го порядка по рекуррентной формуле

$$f(x_i; x_{i+1}; \dots; x_{i+k}) = \frac{f(x_{i+1}; x_{i+2}; \dots; x_{i+k}) - f(x_i; x_{i+1}; \dots; x_{i+k-1})}{x_{i+k} - x_i},$$

например,

$$f(x_0; x_1; \dots; x_n) = \frac{f(x_1; x_2; \dots; x_n) - f(x_0; x_1; \dots; x_{n-1})}{x_n - x_0}.$$

Вычисления разделенных разностей удобно проводить в табл. 5.2 ($n = 4$).

Таблица 5.2

		Разделенные разности			
x_i	$f(x_i)$	1-го порядка	2-го порядка	3-го порядка	4-го порядка
x_0	$f(x_0)$				
x_1	$f(x_1)$	$f(x_0; x_1)$			
x_2	$f(x_2)$	$f(x_1; x_2)$	$f(x_0; x_1; x_2)$		
x_3	$f(x_3)$	$f(x_2; x_3)$	$f(x_1; x_2; x_3)$	$f(x_0; x_1; x_2; x_3)$	
x_4	$f(x_4)$	$f(x_3; x_4)$	$f(x_2; x_3; x_4)$	$f(x_1; x_2; x_3; x_4)$	$f(x_0; x_1; x_2; x_3; x_4)$

Отметим следующие основные свойства разделенных разностей [24].

1. Если функция $f(x)$ n раз непрерывно дифференцируема, то выполняется соотношение

$$f(x_0; x_1; \dots; x_n) = \frac{1}{n!} f^{(n)}(\xi), \text{ где } x_0 < \xi < x_n.$$

2. Разделенные разности k -го порядка являются симметричными функциями своих аргументов и выражаются через узловые значения функции

$$f(x_i; x_{i+1}; \dots; x_{i+k}) = \sum_{j=i}^{i+k} \frac{f(x_j)}{\omega_k^i(x_j)}.$$

3. Разделенные разности n -го порядка от алгебраического многочлена n -й степени постоянны, а разности более высоких порядков равны нулю.

Последнее свойство позволяет выбирать степень интерполяционного полинома такой, чтобы она совпадала с порядком практически постоянных разделенных разностей.

Интерполяционный полином может быть записан через разделенные разности в виде

$$P_n(x) = f(x_0) + (x - x_0) f(x_0; x_1) + (x - x_0)(x - x_1) f(x_0; x_1; x_2) + \cdots + (x - x_0)(x - x_1) \cdots (x - x_{n-1}) f(x_0; x_1; \dots; x_n). \quad (5.12)$$

Степень этого полинома не выше n . Кроме того, легко убедиться, что для такого полинома выполняются условия интерполяции (5.4). Полином, записанный в виде (5.12), называется *интерполяционным полиномом Ньютона для неравноточащих узлов*. Полиномы n -й степени, определяемые формулами (5.8) и (5.12), тождественно равны, так как принимают равные значения в $n+1$ -й точке и представляют собой различные формы записи одного и того же интерполяционного полинома.

Форма записи (5.12) удобна тем, что при добавлении к узлам x_0, x_1, \dots, x_n нового узла x_{n+1} все вычисленные ранее члены остаются без изменения, а в (5.12) добавляется только одно слагаемое.

Отметим, что в (5.12) для вычисления полинома Ньютона используются разделенные разности, расположенные по верхней диагонали (в табл. 5.2 подчеркнуты). Если функция $f(x)$ достаточно гладкая, а узлы интерполяции и точка x принадлежат малому отрезку, то слагаемые в (5.12) убывают. Поэтому на практике вычисления продолжают до тех пор, пока слагаемые остаются больше допустимой погрешности.

Если число узлов задано наперед, то значения полинома Ньютона удобнее вычислять по схеме Горнера, представив его в виде

$$P_n(x) = f(x_0) + (x - x_0) (f(x_0; x_1) + (x - x_1) (f(x_0; x_1; x_2) + \cdots + (x - x_{n-2}) (f(x_0; x_1; \dots; x_{n-1}) + (x - x_{n-1}) f(x_0; x_1; \dots; x_n)) \dots). \quad (5.13)$$

Вычисления по этой формуле на микрокалькуляторе при $n+1=4$ реализуются программой 15, а вычисление значений интерполяционного полинома Ньютона по формуле (5.12) для произвольного числа узлов по составленной таблице раз-

деленных разностей реализуется на микрокалькуляторе с помощью программы 16.

Если в (5.12) положить $n = 1$, то получим часто используемую формулу линейной интерполяции

$$P_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0). \quad (5.14)$$

Вычисления по этой формуле реализуются программой 13.

Запишем выражение для интерполяционного многочлена Ньютона в случае равноотстоящих узлов, если $x_{i+1} - x_i = h = \text{const}$ ($i = 0, 1, \dots, n - 1$). При этом будем использовать конечные разности для табличных значений функции $y_i = f(x_i)$ ($i = 0, 1, \dots, n$).

Конечные разности определяются рекуррентными формулами:

$$\begin{aligned} \Delta y_i &= y_{i+1} - y_i, \quad \Delta^2 y_i = \Delta(\Delta y_i) = \Delta y_{i+1} - \Delta y_i = \\ &= y_i - 2y_{i+1} + y_{i+2}, \dots, \quad \Delta^n y_i = \Delta(\Delta^{n-1} y_i) = \\ &= \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i. \end{aligned}$$

При вычислениях их удобно записывать в виде табл. 5.3 ($n = 4$).

Таблица 5.3

x_i	y_i	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$
x_1	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	
x_2	y_2	Δy_2	$\Delta^2 y_2$		
x_3	y_3	Δy_3			
x_4	y_4				

При $x_i = x_0 + ih$ ($i = 0, 1, \dots, n$) выполняется соотношение между конечными и разделенными разностями [24]

$$f(x_0; x_1; \dots; x_n) = \frac{\Delta^n y_0}{n! h^n}.$$

Для интерполяции функции в точке x , близкой к x_0 , введем новую переменную $t = \frac{x - x_0}{h}$. Тогда узлам интерполяции соответствуют значения $t_0 = 0; t_1 = 1; \dots; t_n = n$.

Множители в (5.12) преобразуются следующим образом:

$$\frac{x - x_0}{h} = t, \quad \frac{x - x_1}{h} = t - 1, \dots, \quad \frac{x - x_{n-1}}{h} = t - n + 1.$$

На основании (5.12) получим формулу интерполяционного полинома Ньютона для интерполяции вперед:

$$P_n(t) = y_0 + \frac{t}{1!} \Delta y_0 + \frac{t(t-1)}{2!} \Delta^2 y_0 + \cdots + \\ + \frac{t(t-1)\cdots(t-n+1)}{n!} \Delta^n y_0. \quad (5.15)$$

Здесь используются конечные разности, расположенные на верхней диагонали в табл. 5.3.

Для интерполяции функции в точке x , близкой к узлу x_n , удобнее ввести переменную $t = \frac{x - x_n}{h}$. В этом случае получим формулу *интерполяционного полинома Ньютона для интерполяции назад*:

$$P_n(t) = y_n + \frac{t}{1!} \Delta y_{n-1} + \frac{t(t+1)}{2!} \Delta^2 y_{n-2} + \cdots + \frac{t(t+1)\cdots(t+n-1)}{n!} \Delta^n y_0. \quad (5.16)$$

В ней используются конечные разности, расположенные на нижней диагонали в табл. 5.3. Вычисление значений полинома Ньютона для равноотстоящих узлов по формулам (5.15) и (5.16) (число узлов $n - 1 \leq 5$) реализуется в автоматическом режиме программой 14.

Интерполяционный многочлен Эрмита. Рассмотрим теперь более общую задачу интерполяирования, когда требуется, чтобы в узлах интерполяции совпадали не только значения искомой функции $f(x)$ и интерполяционной функции $P(x)$, но и значения их производных вплоть до некоторого порядка. Если $P(x)$ — алгебраический многочлен n -й степени, то его называют *интерполяционным многочленом Эрмита* и обозначают $H_n(x)$.

Многочлен $H_n(x)$ определяется из условий

Если в узле x_k ($k = 0, 1, \dots, s$) полином и функция совпадают по производной порядка $\alpha_k - 1$, то число α_k называется кратностью узла x_k . При этом $\alpha_1 + \alpha_2 + \dots + \alpha_s = n + 1$, где n — степень полинома.

Интерполяционная формула Ньютона (5.12) для неравных промежутков обобщается на случай кратных узлов [4]:

$$\begin{aligned}
 H_n(x) = & f(x_0) + (x - x_0) \underbrace{f(x_0; x_0)}_{\alpha_0 \text{ раз}} + (x - x_0)^2 \underbrace{f(x_0; x_0; x_0)}_{\alpha_1 \text{ раз}} + \\
 & + \cdots + (x - x_0)^{\alpha_0-1} \underbrace{f(x_0; \dots; x_0)}_{\alpha_0 \text{ раз}} + (x - x_0)^{\alpha_0} \underbrace{f(x_0; \dots; x_0; x_1)}_{\alpha_1 \text{ раз}} + \\
 & + (x - x_0)^{\alpha_0} (x - x_1) \underbrace{f(x_0; \dots; x_0; x_1; x_1)}_{\alpha_2 \text{ раз}} + \cdots + \\
 & + (x - x_0)^{\alpha_0} (x - x_1)^{\alpha_1} \cdots (x - x_s)^{\alpha_s-1} \underbrace{f(x_0; \dots; x_0; \dots;)}_{\alpha_s \text{ раз}} \\
 & \quad \underbrace{x_s, \dots; x_s}_{\alpha_s \text{ раз}}.
 \end{aligned} \tag{5.18}$$

Разделенные разности в случае кратных узлов вычисляются по формулам:

$$\begin{aligned}
 f(x_0; \dots; x_0) &= \frac{f^{(k-1)}(x_0)}{(k-1)!}, \dots, \\
 f(\underbrace{x_0; \dots; x_0}_{\alpha_0 \text{ раз}}; \underbrace{x_1; \dots; x_1}_{\alpha_1 \text{ раз}}; \dots; \underbrace{x_s; \dots; x_s}_{\alpha_s \text{ раз}}) &= \\
 = \frac{1}{x_s - x_0} (f(\underbrace{x_0; \dots; x_0}_{\alpha_0-1 \text{ раз}}; \underbrace{x_1; \dots; x_1}_{\alpha_1 \text{ раз}}; \dots; \underbrace{x_s; \dots; x_s}_{\alpha_s \text{ раз}}) - \\
 - f(\underbrace{x_0; \dots; x_0}_{\alpha_0 \text{ раз}}; \underbrace{x_1; \dots; x_1}_{\alpha_1 \text{ раз}}; \dots; \underbrace{x_s; \dots; x_s}_{\alpha_s-1 \text{ раз}})). \tag{5.19}
 \end{aligned}$$

Выпишем общее выражение интерполяционного полинома Эрмита [4]

$$\begin{aligned}
 H_n(x) = & \sum_{i=0}^n \sum_{j=0}^{\alpha_i-1} \sum_{k=0}^{\alpha_i-j-1} f^{(j)}(x_i) \frac{1}{k!} \frac{1}{j!} (x - x_i)^{j+k} \times \\
 & \times \prod_{m \neq i} (x - x_m)^{\alpha_m} \left\{ \frac{d^k}{dx^k} \prod_{r \neq i} (x - x_r)^{-\alpha_r} \Big|_{x=x_i} \right\}.
 \end{aligned} \tag{5.20}$$

Формулы (5.18) и (5.20) в общем случае достаточно громоздки. Чем более высокие кратности узлов у полинома Эрмита заданной степени, тем меньше требуется брать узлов, что уменьшает вычислительную погрешность. На практике обычно выбирают немного узлов с небольшой кратностью. При этом легко составить таблицу разделенных разностей с кратными узлами. В первый столбец записываются узлы интерполя-

ции, причем каждый узел повторяется столько раз, какова его кратность. Во второй столбец выписываются соответствующие узлам значения функции. В третий столбец помещают разделенные разности 1-го порядка, если соответствующие аргументы не совпадают, или первые производные, если аргументы совпадают. В четвертый столбец помещают разделенные разности 2-го порядка, если они вычисляются так же, как обычные разделенные разности, или же производные второго порядка, деленные на $2!$, если аргументы, на разность которых нужно делить, совпадают, и т. д. В частности, для двух узлов x_0 и x_1 кратности $\alpha_0 = 2$ и $\alpha_1 = 2$ получим табл. 5.4.

Таблица 5.4

x_i	y_i	Разделенные разности с кратными узлами		
		1-го порядка	2-го порядка	3-го порядка
x_0	y_0			
x_0	y_0	y'_0		
x_1	y_1	$\frac{y_1 - y_0}{x_1 - x_0}$	$\frac{y_1 - y_0}{(x_1 - x_0)^2} - \frac{y'_0}{x_1 - x_0}$	
x_1	y_1	y'_1	$\frac{y'_1}{x_1 - x_0} - \frac{y_1 - y_0}{(x_1 - x_0)^2}$	$\frac{y'_1 + y'_0}{(x_1 - x_0)^2} - \frac{2(y_1 - y_0)}{(x_1 - x_0)^3}$

Разности, стоящие по диагонали табл. 5.4, являются коэффициентами многочлена $H_n(x)$. Для указанного частного случая выпишем интерполяционный многочлен Эрмита

$$H_3(x) = y_0 + (x - x_0)y'_0 + (x - x_0)^2 \left(\frac{y_1 - y_0}{(x_1 - x_0)^2} - \frac{y'_0}{x_1 - x_0} \right) + \\ + (x - x_0)^2 (x - x_1) \left(\frac{y'_1 + y'_0}{(x_1 - x_0)^2} - \frac{2(y_1 - y_0)}{(x_1 - x_0)^3} \right). \quad (5.21)$$

Для интерполяционного многочлена Эрмита оценка погрешности [4]

$$|f(x) - H_n(x)| = \frac{|f^{(n+1)}(\xi)|}{(n+1)!} |(x - x_0)^{\alpha_0} (x - x_1)^{\alpha_1} \dots \\ \dots (x - x_s)^{\alpha_s}|, \quad (5.22)$$

где ξ — внутренняя точка минимального отрезка, содержащего узлы интерполяции x_0, x_1, \dots, x_s и точку x . Отметим, что на одной и той же сетке узлов численная величина погрешности полинома Ньютона будет больше, чем у полинома Эрмита.

5.3. Лабораторная работа 5. Полиномиальная интерполяция

Программа 11 вычисления интерполяционного полинома Лагранжа по формуле (5.8) для числа узлов $n + 1 \leq 9$ в автоматическом режиме:

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ПД	4Г	15	F,	25	30	—	11
01	ПО	40	16	С/П	50	31	×	12
02	ПС	4[17	БП	51	32	FL0	5Г
03	С/П	50	18	22	22	33	22	22
04	ПА	4—	19	ИПО	60	34	+	10
05	С/П	50	20	ПС	4[35	БП	51
06	КП↑	ЛЕ	21	F,	25	36	10	10
07	FL0	5Г	22	КИП↑	ГЕ	37	F,	25
08	05	05	23	ИПВ	6Л	38	FL0	5Г
09	0	00	24	—	11	39	19	19
10	ИПД	6Г	25	FX≠0	57	40	+	10
11	ПО	40	26	37	37	41	БП	51
12	F,	25	27	÷	13	42	10	10
13	КИПС	Г[28	КИП↑	ГЕ	6—		
14	ПВ	4Л	29	ИПА				

Распределение регистров памяти:

Номер регистра	0	1, ..., 9	A	B	C	D
Переменная	счетчик цикла	x_n, \dots, x_0	x	x_k	k	$n + 1$

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В режиме автоматической работы (F АВТ) вводить исходные данные и осуществлять вычисления по схеме: $(n + 1)$ В/О С/П (x) С/П (x_0) С/П (x_1) С/П ... (x_n) С/П (y_0) С/П (y_1) С/П ... (y_n) С/П (каждое очередное число вводить после ОСТАНОВа). После последнего ОСТАНОВа списываем с индикатора значение $P_n(x)$.

Контрольный пример. Вычислить значение интерполяционного полинома Лагранжа в точке $x = 0,17$ для функции, заданной в пяти узлах таблицей:

x_i	0	0,1	0,2	0,3	0,4
y_i	1,00	1,11	1,22	1,35	1,49

- Нажмем клавиши F ПРГ и клавиши программы.
- Нажмем клавиши F АВТ и приступим к вводу данных и вычислениям:

5 В/О С/П 0,17 С/П 0 С/П 0,1 С/П 0,2 С/П 0,3 С/П
0,4 С/П 1 С/П 1,11 С/П 1,22 С/П 1,35 С/П 1,49 С/П

После ОСТАНОВа прочтем на индикаторе 1.1852299. Ответ:
 $P_4(0,17) = 1,19$. Время ввода данных и счета около 2 мин.

Программа 12 выполнения одного шага интерполяционной схемы Эйткена для вычисления значений полинома Лагранжа. Используется для вычисления элементов столбцов табл. 5.1 по формуле

$$L_{i-k, \dots, i}(x) = \frac{1}{x_i - x_{i-k}} \begin{vmatrix} L_{i-k, \dots, i-1}(x) & x_{i-k} - x \\ L_{i-k+1, \dots, i}(x) & x_i - x \end{vmatrix}.$$

При этом $L_{0,1, \dots, n}(x) = P_n(x)$. Новые узлы интерполяции можно подключать в процессе счета.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	6	06	05	—	11	10	ИП4	64
01	ПО	40	06	×	12	11	ИП1	61
02	КИПО	Г0	07	FL0	5Г	12	—	11
03	КИПО	Г0	08	02	02	13	÷	13
04	ИП6	66	09	—	11	14	С/П	50

Распределение регистров памяти:

Номер регистра	1	2	4	5	6
Переменная	x_{i-k}	$L_{i-k+1, \dots, i}$	x_i	$L_{i-k, \dots, i-1}$	x

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) ввести исходные данные:

(x) П6 (x_{i-k}) П1 (x_i) П4 ($L_{i-k}, \dots, i-1$) П5 (L_{i-k+1}, \dots, i) П2.

3. Осуществить пуск программы: В/О С/П. После ОСТАНОВа списать с индикатора значение L_{i-k}, \dots, i (x) и заполнить таблицу. На диагонали получим значения полинома Лагранжа соответствующей степени (равной номеру узла) в точке x .

Контрольный пример. Применяя схему Эйткена, вычислить значения интерполяционного полинома Лагранжа степени $n = 1, 2, 3$ в точке $x = 1$ для функции, заданной таблицей:

x_i	y_i	$L_{i-1, i}$	$L_{i-2, i-1, i}$	$L_{i-3, i-2, i-1, i}$
0	1,0			
1,5	4,5	3,33333		
2	7,4	1,6	2,46667	
3	20,1	-5,3	3,9	2,94444

После ввода программы в автоматическом режиме вводим исходные данные:

1 П6 0 П1 1,5 П4 1 П5 4,5 П2 В/О С/П.

После останова запишем в таблицу 3,33333 и запесем это значение в РА, нажав клавиши ПА:

1,5 П1 2 П4 4,5 П5 7,4 П2 В/О С/П

Значение 1,6 запишем в таблицу и зашлем в РВ: ПВ.

2 П1 3 П4 7,4 П5 20,1 П2 В/О С/П

Значение — 5,3 запишем в таблицу и зашлем в РС: ПС.

0 П1 2 П4 ИПА П5 ИПВ П2 В/О С/П

Значение 2,46667 запишем в таблицу и зашлем в РА: ПА.

1,5 П1 3 П4 ИПВ П5 ИПС П2 В/О С/П

Значение 3,9 запишем в таблицу и зашлем в Р2: П2.

0 П1 3 П4 ИПА П5 В/О С/П

Значение 2,94444 запишем в таблицу.

Ответ: $P_1(1) = 3,33333$, $P_2(1) = 2,46667$, $P_3(1) = 2,94444$.

Время вычислений одного элемента таблицы — 5 с.

Программа 13 вычисления в автоматическом режиме значений функции, заданной в двух точках, по формуле линейной интерполяции (5.14).

Адресс	Команда	Код	Адресс	Команда	Код
00	П1	41	10	С/П	50
01	С/П	50	11	ИП1	61
02	—	11	12	—	11
03	С/П	50	13	ИП3	63
04	П2	42	14	×	12
05	С/П	50	15	ИП2	62
06	—	11	16	+	10
07	Х У	14	17	БП	51
08	÷	13	18	10	10
09	Г3	43			

Распределение регистров памяти:

Номер регистра	1	2	3
Переменная	x_0	y_0	Δy_0

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В режиме автоматической работы (F АВТ) вводить исходные данные и осуществлять вычисления по схеме: (x_0) В/О С/П (x_1) С/П (y_0) С/П (y_1) С/П (x) С/П, прочесть на индикаторе значение $P_1(x)$.

3. Для вычисления значений этой же функции в другой точке x достаточно нажать клавиши (x) С/П.

Контрольный пример. По таблице значений функции

x_i	1,517	2,314
y_i	7,211	3,151

вычислить значение $P_1(x)$ при $x = 1,943$.

Вводим исходные данные: 1,517 В/О С/П 2,314 С/П 7,211 С/П 3,151 С/П 1,943 С/П.

Ответ: $P_1(1,943) = 5,041$. Время вычислений — 3 с.

Программа 14 вычисления в автоматическом режиме значений интерполяционного полинома Ньютона для равноотстоящих узлов по формулам (5.15) и (5.16) при $n + 1 \leq 5$:

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИПА	6—	12	ПИ	41	24	ИП6	66
01	—	11	13	ИПД	6Г	25	ИП0	60
02	ИПС	6+	14	+ ПД	10	26	—	11
03	+	10	15	ПД	4Г	27	—	11
04	ИПС	6/	16	ИП6	66	28	ВХ	0
05	÷	13	17	ПО	40	29	÷	13
06	ПВ	4Л	18	ИПД	6Г	30	×	12
07	1	01	19	С/П	50	31	КП↑	Л Е
08	Г6	46	20	КП6	Л 6	32	FL0	5Г
09	0	00	21	КИП†	ГЕ	33	21	21
10	ПД	4Г	22	—	11	34	БП	51
11	С/П	50	23	ИПВ	6Л	35	13	13

Распределение регистров памяти:

Номер регистра	0	1, ..., 5	6	А	В	С	Д
Переменная	счетчик цикла	y_0, \dots, y_4	регистр косвен. адресации	x_0	$\frac{x - x_i}{h}$	h	$P_n(x)$

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) ввести данные и приступить к вычислениям по схеме:

(x_0) ПА (h) ПС (x) В/О С/П.

3. После каждого ОСТАНОВа вводим последовательно значения функции (y_0) С/П (y_1) С/П При этом на индикаторе высвечивается значение полинома $P_i(x)$, вычисленное по введенным значениям функции. Исчерпав все точки (или достигнув заданной точности), запишем окончательное значение $P_n(x)$.

Для интерполяции назад (формула (5.16)) можно занести в регистр С значение $(-h)$, при этом, естественно, вместо x_0 заносим последний узел и ввод y_k начинаем с y_n .

Контрольный пример. По таблице значений $f(x)$

x_i	0	0,25	0,50	0,75	1
y_i	1	1,31	1,75	2,31	3,00

вычислить значение $P_4(x)$ при $x = 0,2$.

Ответ: $P_4(0,2) = 1,24$. Время ввода и счета равно 1,5 мин.

Программа 15 вычисления в автоматическом режиме значений интерполяционного полинома Ньютона для неравнотстоящих узлов ($n + 1 = 4$) по формуле (5.13):

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИП2	62	17	КИП4	Г4	34	П0	40
01	П3	43	18	КИП4	Г4	35	6	06
02	1	01	19	XV	14	36	П4	44
03	0	00	20	BХ	0	37	1	01
04	П0	40	21	—	11	38	4	04
05	6	06	22	КИП0	Г0	39	П1	41
06	ИП3	63	23	КИП1	Г1	40	КИП1	Г1
07	—	10	24	—	11	41	ИП5	65
08	П1	41	25	÷	13	42	КИП4	Г4
09	1	01	26	КП5	Л5	43	—	11
10	2	02	27	XV	14	44	×	12
11	ИП3	63	28	FL3	5—	45	КИП1	Г1
12	—	11	29	18	18	46	+	10
13	П4	44	30	FL2	58	47	FL0	5Г
14	1	01	31	00	00	48	41	41
15	—	10	32	C/П	50	49	C/П	50
16	П5	45	33	3	03			

Распределение регистров памяти:

Номер регистра	0, ..., 4	5	6	7	8	9	А	В	С	Д
Переменная	счетчики циклов	x	x_3	x_2	x_1	x_0	y_0	y_1	y_2	y_3

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) ввести исходные данные: узлы интерполяции x_0, x_1, x_2, x_3 соответственно в регистры 9, 8, 7 и 6, значения функции y_0, y_1, y_2, y_3 соответственно в регистры А, В, С и Д, 3 в Р2.

3. Нажать клавиши В/О С/П.

4. После ОСТАНОВа (через 55 с) ввести значение x в Р5. Нажать клавишу С/П. После следующего ОСТАНОВа (через 15 с) прочесть на индикаторе $P_3(x)$.

5. Если по тем же узлам интерполяции требуется вычислить значение полинома Ньютона в другой точке x , то вводим: (x) П5 БП 33 С/П.

Контрольный пример. Для таблицы значений функции

x_i	0	1	2	3
y_i	0	0,5	0,866	1,0

найти значение интерполяционного полинома Ньютона третьей степени при $x = 0,7$. После ввода программы вводим исходные данные:

0 П9 1 П8 2 П7 3 П6 0 ПА 0,5 ПВ 0,866 ПС 1 ПД 3 П2

Нажимаем клавиши В/О, С/П, после ОСТАНОВа вводим значение аргумента $x = 0,7$ в Р5, С/П. После ОСТАНОВа получим на индикаторе $P_3(0,7) = 0,3596$.

Программа 16 вычисления значений в полуавтоматическом режиме интерполяционного полинома Ньютона по формуле (5.12) для таблицы разделенных разностей функции с любым числом узлов:

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИП1	61	08	ИП1	61	16	Х	12
01	ИПА	6—	09	ХУ	14	17	ИПД	6Г
02	—	11	10	—	11	18	+	10
03	ПС	4[11	ИПС	6[19	ПД	4Г
04	С/П	50	12	Х	12	20	БП	51
05	ПД	4Г	13	ПС	4[21	06	06
06	С/П	50	14	ВХ	0			
07	ПВ	4L	15	С/П	50			

Распределение регистров памяти:

Номер регистра	1	A	B	C	D
Переменная	x	x_0	x_i	$\prod_{k=0}^i (x - x_k)$	$P_i(x)$

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) ввести исходные данные: (x) П1 (x_0) ПА и нажать клавиши В/О С/П. При первом ОСТАНОВе набрать на клавиатуре значение y_0 и нажать клавишу С/П.

3. При последующих ОСТАНОВах вводить последовательно значения узлов (x_i) С/П, значения конечных разностей ($f(x_0; x_1; \dots; x_i)$) (стоящих на верхней диагонали табл. 5.2) С/П, после ОСТАНОВа на индикаторе прочесть значение полинома $P_i(x)$.

4. Повторить п. 3 для каждого узла интерполяции, записывая, если нужно, значения полиномов Ньютона соответствующей степени в точке x .

Для новой таблицы значений функции повторить п. 2—4. Предполагается, что для значений функции составлена таблица разделенных разностей (см. табл. 5.2).

Контрольный пример. Для функции, заданной таблицей, вычислить значение полинома Ньютона при $x = 0,6$:

x_i	y_i	Разделенные разности				
0	1,0					
0,2	1,2	1,00				
0,5	1,6	1,33	0,66			
0,7	2,0	2,00	1,34	0,97		
1,0	2,7	2,33	0,66	-0,85	-1,82	

После ввода программы запосим исходные данные: 0,6 П1 0 ПА В/О С/П; после ОСТАНОВа 1 С/П; после ОСТАНОВа 0,2 С/П, 1 С/П, прочтем на индикаторе значение $P_1(0,6) = 1,60$; 0,5 С/П 0,66 С/П, прочтем значение $P_2(0,6) = 1,76$; 0,7 С/П 0,97 С/П, прочтем значение $P_3(0,6) = 1,78$; 1 С/П 1,82 /— С/П, прочтем значение $P_4(0,6) = 1,79$ и т. д., при необходимости в процессе вычислений можно добавлять новые узлы интерполяции.

Задания к лабораторной работе 5. Для функции, заданной таблицей

x_i	0	1,25	0,50	0,75	1,00
y_i	1,00	1,28	1,65	2,12	2,72

вычислить значение полинома Лагранжа в точке $x = 1 - 0,033n$ и полинома Ньютона в точке $x = 0,033n$ ($n = 1, \dots, 30$ — номер варианта). Для вариантов 1—15 использовать формулу Ньютона интерполяции вперед (5.15), для вариантов 16—30 — формулу интерполяции назад (5.16).

5.4. Аппроксимация функции методом наименьших квадратов

Рассмотренные методы интерполяции обеспечивают аналитическое представление (или вычисление) таблично заданной функции $f(x)$ лишь на небольшом интервале с несколькими узлами интерполяции. Если требуется получить аппроксимирующую функцию $\varphi(x)$ при значительном числе наблюдений $f(x_i)$, то повышение степени интерполяционного полинома усложняет вычисления и увеличивает погрешность.

Кроме того, процесс вычисления разностей, используемый в интерполяции, является неустойчивым, т. е. даже незначительные ошибки в значениях функции приводят к тому, что разности высоких порядков практически носят случайный характер.

Поэтому, если значения функции $f(x)$ измерены в результате эксперимента со случайными погрешностями, называемыми в технике шумами, то проводить интерполяцию нецелесообразно. В таких случаях разумнее применять средние квадратические приближения, так как они сглаживают отдельные локальные ошибки наблюдений и лучше представляют реальную функциональную зависимость. Построение такого приближения называют сглаживанием экспериментальных данных.

Наиболее эффективным методом построения наилучшего среднего квадратического приближения является метод наименьших квадратов.

Пусть функция $f(x)$ задана значениями, которые могут содержать случайные ошибки, на дискретном или непрерывном множестве $X \subseteq [a; b]$. Выберем систему линейно независимых на X функций $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$ и рассмотрим их всевозможные линейные комбинации $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$.

Множество обобщенных полиномов $\varphi(x)$ образует $(n+1)$ -мерное линейное подпространство H_{n+1} . Будем аппроксимировать функцию $f(x) \notin H_{n+1}$ обобщенным полиномом $\varphi(x)$.

По принципу наименьших квадратов параметры c_0, c_1, \dots, c_n аппроксимирующего обобщенного полинома $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$ определяются из условия, чтобы квадрат взвешенного среднего квадратического отклонения функций $f(x)$ и $\varphi(x)$ на множестве X :

$$\rho^2(f, \varphi) = \delta_n^2 \equiv \Phi(c_0, c_1, \dots, c_n) \equiv \|f - \varphi\|^2 = (f - \varphi, f - \varphi) = (f, f) - 2 \sum_{i=0}^n c_i (\varphi_i, f) + \sum_{i=0}^n \sum_{j=0}^n c_i c_j (\varphi_i, \varphi_j) \quad (5.24)$$

был наименьшим. Многочлен $\varphi(x)$, минимизирующий $\rho^2(f, \varphi)$, называется многочленом *наилучшего среднего квадратического приближения* функции $f(x)$, а метод его отыскания — *методом наименьших квадратов*.

Для определения минимума функции $\Phi(c_0, c_1, \dots, c_n)$ приравниваем к нулю ее частные производные по неизвестным параметрам

$$\frac{1}{2} \frac{\partial \Phi}{\partial c_k} = \sum_{i=0}^n c_i (\varphi_k, \varphi_i) - (f, \varphi_k) = 0 \quad (5.25)$$

(k = 0, 1, 2, \dots, n).

Получим систему линейных алгебраических уравнений в развернутом виде

Систему (5.26) называют *нормальной* системой метода наименьших квадратов. Определитель системы есть *определитель Грама* и в случае линейно независимых функций $\varphi_i(x)$ ($i = 0, 1, \dots, n$) отличен от нуля, т. е. коэффициенты c_i определяются однозначно.

Отметим, что функция $\Phi(c_0, c_1, \dots, c_n)$ в (5.24) есть квадратичная форма относительно аргументов, поэтому она имеет единственную стационарную точку, определяемую необходимыми условиями (5.25). Ввиду положительности квадратичной формы решение системы (5.26) определяет минимум этой функции.

Для решения нормальной системы уравнений (5.26) можно применять известные методы (см. гл. 3), например метод Гаусса—Жордана. Однако, если размерность системы велика, то ее матрица плохо обусловлена и параметры аппроксимации определяются с большими погрешностями. Поэтому изложенную выше методику обычно применяют в тех случаях, когда число параметров невелико. Во всех остальных случаях лучше использовать приближение ортогональными функциями.

Параметры аппроксимации c_i ($i = 0, 1, \dots, n$) наиболее просто определяются в том частном случае, когда базисные

функции $\varphi_i(x)$ ($i = 0, 1, \dots, n$) ортогональны на множестве X , т. е.

$$(\varphi_i, \varphi_k) = 0 \quad (i \neq k). \quad (5.27)$$

При таком выборе базисных функций матрица системы уравнений (5.26) становится диагональной, а параметры

$$c_i = \frac{(f, \varphi_i)}{(\varphi_i, \varphi_i)} \quad (i = 0, 1, \dots, n) \quad (5.28)$$

называются *обобщенными коэффициентами Фурье* функции $f(x)$ относительно системы ортогональных функций $\{\varphi_i(x)\}_0^n$.

Выбор ортогонального базиса удобен тем, что при увеличении порядка обобщенного полинома значения всех предыдущих коэффициентов c_0, c_1, \dots, c_n не изменяются, в то время как в любой неортогональной системе функций все вычисления метода наименьших квадратов нужно проводить заново.

В случае ортогонального базиса выражение (5.24) для квадрата отклонения функции $\varphi(x)$ от $f(x)$ имеет достаточно простой вид:

$$\begin{aligned} \delta_n^2 &= \|f - \varphi\|^2 = (f, f) - \sum_{i=0}^n c_i^2 (\varphi_i, \varphi_i) = \\ &= \|f\|^2 - \sum_{i=0}^n c_i^2 \|\varphi_i\|^2. \end{aligned} \quad (5.29)$$

Поскольку $\delta_n^2 \geq 0$, то из (5.29) следует неравенство:

$$\sum_{i=0}^n c_i^2 \|\varphi_i\|^2 \leq \|f\|^2. \quad (5.30)$$

Если функции $\{\varphi_i(x)\}_0^n$ ортонормированы, то для коэффициентов Фурье выполняется *неравенство Бесселя*:

$$\sum_{i=0}^n c_i^2 \leq \|f\|^2. \quad (5.31)$$

При увеличении порядка n обобщенного полинома $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$ погрешность среднего квадратического приближения δ_n^2 в силу равенства (5.29) убывает. Если

$$\lim_{n \rightarrow \infty} \|f - \sum_{i=0}^n c_i \varphi_i\|_{L_2[a, b]} = 0, \quad (5.32)$$

то говорят, что последовательность обобщенных полиномов по ортогональной системе сходится в среднем к функции

$f(x)$. Сходимость в среднем связана с понятием полноты системы базисных функций (см. п. 5.1).

Если система функций $\varphi_0(x), \varphi_1(x), \varphi_2(x), \dots$ является полной в линейном нормированном пространстве L , то для любой функции $f(x) \in L$ и любого $\epsilon > 0$ существует обобщенный полином n -го порядка $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x) \in H_{n+1} \subset L$ такой,

что $\|f - \varphi\| < \epsilon$. В этом случае ряд $\sum_{i=0}^{\infty} c_i \varphi_i(x)$ сходится к функции $f(x)$ в метрике L , т. е. для n -й частичной суммы $S_n(x) = \varphi(x)$ выполняется соотношение

$$\lim_{n \rightarrow \infty} \|f - S_n\| = 0. \quad (5.33)$$

В частности, обобщенный ряд Фурье $\sum_{i=0}^{\infty} c_i \varphi_i(x)$ для произвольной функции $f(x) \in L_2^p[a; b]$ по полной ортонормированной системе функций $\varphi_0(x), \varphi_1(x), \varphi_2(x), \dots$ сходится в среднем к $f(x)$. Условие полноты ортонормированной системы функций эквивалентно тому, что для любой функции $f(x) \in L_2^p[a; b]$ выполняется равенство Парсеваля

$$\sum_{i=0}^{\infty} c_i^2 = \|f\|^2, \quad (5.34)$$

получаемое из неравенства Бесселя (5.31) при $n \rightarrow \infty$, где c_i — коэффициенты Фурье функции $f(x)$ по данной системе функций. Равенство (5.34) называют еще *уравнением замкнутости*. В пространстве $L_2^p[a; b]$ понятия замкнутости и полноты эквивалентны.

Если X — дискретное множество точек x_0, x_1, \dots, x_m , то любая система $(m + 1)$ -й функций, линейно независимых на множестве точек X , будет полной на этом множестве.

Рассмотрим алгоритмы аппроксимации функции $f(x)$, заданной на дискретном или непрерывном множестве $X \subseteq [a; b]$, методом наименьших квадратов при различном выборе базисных функций $\varphi_i(x)$ ($i = 0, 1, \dots, n$).

5.5. Точечная аппроксимация функции методом наименьших квадратов

Пусть в результате наблюдений на дискретном множестве точек x_0, x_1, \dots, x_m получена таблица значений функции $f(x)$: $y_0 = f(x_0), y_1 = f(x_1), \dots, y_m = f(x_m)$.

Обычно предполагается, что значения аргумента определены значительно точнее, чем значения функции. Построим на указанном дискретном множестве точек среднее квадратическое приближение в виде обобщенного полинома

$$\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x).$$

По принципу наименьших квадратов требуется, чтобы взвешенная сумма квадратов отклонений значений полинома $\varphi(x)$ от заданных значений функции $f(x)$ во всех узлах x_i ($i = 0, 1, \dots, m$) была минимальной.

Параметры аппроксимирующего полинома $\varphi(x)$, с учетом обозначений (5.3) и (5.24), определяются из условия минимума функции

$$\Phi(c_0, c_1, \dots, c_n) = \sum_{i=0}^m p_i (\varphi(x_i) - f(x_i))^2$$

и являются решением системы линейных алгебраических уравнений

$$\begin{aligned} \sum_{i=0}^m p_i [c_0 \varphi_0(x_i) + c_1 \varphi_1(x_i) + \dots + c_n \varphi_n(x_i)] \varphi_k(x_i) = \\ = \sum_{i=0}^m p_i f(x_i) \varphi_k(x_i) \quad (k = 0, 1, 2, \dots, n). \end{aligned} \quad (5.35)$$

Замечание. Если в методе наименьших квадратов $n = m$, то точечная средняя квадратическая аппроксимация алгебраическим многочленом совпадает с лагранжевой интерполяцией. На практике обычно число искомых параметров выбирают значительно меньшим числа наблюдений: $n \ll m$. Тогда случайные ошибки отдельных измерений погашают друг друга и решение становится более достоверным.

Выбор базисных функций $\varphi_i(x)$ ($i = 0, 1, \dots, n$) обычно диктуется физическим смыслом задачи, при анализе которой получены экспериментальные данные. Иногда одни и те же исходные данные почти с одинаковой точностью удается сгладить кривыми, совершившими различными по функциональной природе.

Рассмотрим простейший случай точечной аппроксимации таблично заданной функции методом наименьших квадратов с помощью обобщенного полинома

$$\varphi(x) = c_0 \varphi_0(x) + c_1 \varphi_1(x), \quad (5.36)$$

зависящего от двух параметров c_0 и c_1 ; $\varphi_0(x)$ и $\varphi_1(x)$ — любые функции, линейно независимые на множестве точек x_0 ,

x_1, \dots, x_m . Пусть веса $p_i = 1$ ($i = 0, 1, \dots, m$). Неизвестные параметры c_0 и c_1 , согласно (5.35), определяются из системы

$$\begin{cases} c_0 \sum_{i=0}^m \varphi_0^2(x_i) + c_1 \sum_{i=0}^m \varphi_0(x_i) \varphi_1(x_i) = \sum_{i=0}^m f(x_i) \varphi_0(x_i); \\ c_0 \sum_{i=0}^m \varphi_0(x_i) \varphi_1(x_i) + c_1 \sum_{i=0}^m \varphi_1^2(x_i) = \sum_{i=0}^m f(x_i) \varphi_1(x_i). \end{cases} \quad (5.37)$$

Решение этой системы по формулам Крамера

$$c_0 = \frac{\sum f(x_i) \varphi_0(x_i) \sum \varphi_1^2(x_i) - \sum f(x_i) \varphi_1(x_i) \sum \varphi_0(x_i) \varphi_1(x_i)}{\sum \varphi_0^2(x_i) \sum \varphi_1^2(x_i) - (\sum \varphi_0(x_i) \varphi_1(x_i))^2};$$

$$c_1 = \frac{\sum \varphi_0^2(x_i) \sum f(x_i) \varphi_1(x_i) - \sum f(x_i) \varphi_0(x_i) \sum \varphi_0(x_i) \varphi_1(x_i)}{\sum \varphi_0^2(x_i) \sum \varphi_1^2(x_i) - (\sum \varphi_0(x_i) \varphi_1(x_i))^2}, \quad (5.38)$$

где суммирование проводится по $i = 0, 1, \dots, m$.

Вычисления по формулам (5.38) реализуются на микрокалькуляторе программой 17.

Программа 17 вычисления параметров аппроксимирующей функции $\varphi(x) = c_0 \varphi_0(x) + c_1 \varphi_1(x)$ методом наименьших квадратов:

6	ПО	П4	П6	С/П	ПП	47	ИП1	КИПО	X
BХ	ИП3	×	КИП4	+	КП6	ХУ	КИП4	+	КП6
FL0	07	FL2	00	ИП7	ИП8	ПП	41	ИП9	ИПА
ПП	41	÷	↑	ИП8	ИПС	÷	ХУ	ПП	41
C/П	ИПВ	×	ИПС	÷	—	B/0	...		

(Подпрограмма вычисления $\varphi_0(x)$ и $\varphi_1(x)$) В/0.

Распределение регистров памяти:

Номер регистра	Д	5	1	3	C	A, B
Переменная	x_i	y_i	$\varphi_0(x_i)$	$\varphi_1(x_i)$	$\sum \varphi_0^2(x_i)$	$\sum \varphi_0(x_i) \varphi_1(x_i)$
Номер регистра	9		7		0,2	4,6
Переменная	$\sum \varphi_1^2(x_i)$	$\sum y_i \varphi_1(x_i)$		Счетчики циклов	Регистры косвенной адресации	

Инструкция. 1. Составить подпрограмму вычисления значений функций $\varphi_0(x)$ и $\varphi_1(x)$. При этом значение x хранится

в РД, а значения функций заносятся соответственно в Р1 и Р3.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В автоматическом режиме (F АВТ) занести число узлов в Р2 и очистить Р7—РС:

($m + 1$) П2 0 П7 П8 П9 ПА ПВ ПС

4. Нажать клавиши В/О С/П. После ОСТАНОВа ввести x_1 и y_1 по схеме: (x_1) ПД (y_1) П5 С/П. После каждого следующего останова вводить данные для остальных узлов:

(x_i) ПД (y_i) П5 С/П.

Перебрав все узлы, на индикаторе получим значение c_0 . Нажав клавишу ХУ, получим значение c_1 .

Контрольный пример. Для функции $f(x)$

x_i	0	0,25	0,5	0,75	1
$y_i = f(x_i)$	0,5	0,9	0,8	0,3	-0,5

определить параметры c_0 и c_1 аппроксимирующей функции вида $\Phi(x) = c_0 \cos \pi x + c_1 \sin \pi x$. Составим подпрограмму вычисления значений функций $\varphi_0(x) = \cos \pi x$ и $\varphi_1(x) = \sin \pi x$, начиная с адреса 47:

ИПД π × cos П1 BX sin П3 В/О

После ввода программы в автоматическом режиме вводим в регистр 2 число узлов 5 П2 и очищаем регистры 7, 8, ..., С: 0 П7 П8 П9 ПА ПВ ПС.

Осуществляем вычисления: В/О С/П

0 ПД 0,5 П5 С/П 0,25 ПД 0,9 П5 С/П
0,5 ПД 0,8 П5 С/П 0,75 ПД 0,3 П5 С/П
1 ПД 0,5 /-/ П5 С/П

Запишем с индикатора значение $c_0 = 0,47475465$. Нажимаем клавишу ХУ, запишем $c_1 = 0,82426405$.

Ответ: $\varphi(x) = 0,47 \cos \pi x + 0,82 \sin \pi x$.

Время счета с вводом исходных данных — около 3 мин.

5.6. Точечная аппроксимация функций алгебраическими многочленами

В приложениях метода наименьших квадратов наиболее часто полагают $\varphi_i(x) = x^i$ ($i = 0, 1, \dots, n$), т. е. в качестве приближающей функции берется многочлен

$$\varphi(x) = c_0 + c_1 x + \dots + c_n x^n.$$

Как отмечалось, степень многочлена обычно выбирается значительно меньше числа наблюдений: $n \ll m$.

Для указанного частного случая полиномиальной аппроксимации коэффициенты и правые части нормальной системы (5.26) определяются по формулам:

$$(\varphi_k, \varphi_l) = \sum_{i=0}^m x_i^{k+l}; (f, \varphi_l) = \sum_{i=0}^m y_i x_i^l (k, l = 0, 1, \dots, n), \quad (5.39)$$

где $m + 1$ — число узлов аппроксимации.

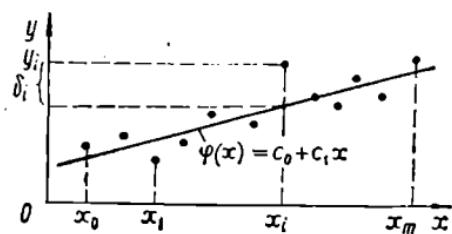


Рис. 12

Если опытные данные при их изображении на графике выстраиваются почти по прямой (рис. 12), то сглаживание удобно проводить прямой, называемой линией регрессии ($n = 1$):

$$\varphi(x) = c_0 + c_1 x.$$

Параметры ее определяются из системы

$$\begin{cases} c_0(m+1) + c_1 \sum_{i=0}^m x_i = \sum_{i=0}^m y_i; \\ c_0 \sum_{i=0}^m x_i + c_1 \sum_{i=0}^m x_i^2 = \sum_{i=0}^m x_i y_i. \end{cases} \quad (5.40)$$

При $n = 2$ параметры квадратичной аппроксимирующей функции

$$\varphi(x) = c_0 + c_1 x + c_2 x^2$$

определяются из системы

$$\begin{cases} c_0(m+1) + c_1 \sum_{i=0}^m x_i + c_2 \sum_{i=0}^m x_i^2 = \sum_{i=0}^m y_i; \\ c_0 \sum_{i=0}^m x_i + c_1 \sum_{i=0}^m x_i^2 + c_2 \sum_{i=0}^m x_i^3 = \sum_{i=0}^m x_i y_i; \\ c_0 \sum_{i=0}^m x_i^2 + c_1 \sum_{i=0}^m x_i^3 + c_2 \sum_{i=0}^m x_i^4 = \sum_{i=0}^m x_i^2 y_i. \end{cases} \quad (5.41)$$

Полученные системы линейных алгебраических уравнений можно решить, например, методом Гаусса—Жордана. На практике применяют полиномы сравнительно невысокой степени ($n \leq 5$), так как при более высоких степенях система нормальных уравнений может оказаться плохо обусловленной. Погрешность аппроксимации линейным и квадратичным

многочленом согласно (5.24) определяется соответственно выражениями:

$$\delta_1 = \left(\sum_{i=0}^m (c_0 + c_1 x_i - y_i)^2 \right)^{1/2}, \quad (5.42)$$

$$\delta_2 = \left(\sum_{i=0}^m (c_0 + c_1 x_i + c_2 x_i^2 - y_i)^2 \right)^{1/2}. \quad (5.43)$$

5.7. Лабораторная работа 6. Точечная аппроксимация функции алгебраическими многочленами методом наименьших квадратов

Для функции $f(x)$, заданной значениями y_0, y_1, \dots, y_m на дискретном множестве точек x_0, x_1, \dots, x_m , вычисляются параметры линейной и квадратичной аппроксимирующих функций методом наименьших квадратов.

Программа 18 вычисления параметров линейной регрессии $\varphi(x) = c_0 + c_1 x$ методом наименьших квадратов из системы (5.40)

Адресс	Команда	Код	Адресс	Команда	Код	Адресс	Команда	Код
00	П2	42	16	+	10	32	×	12
01	Х	12	17	П7	47	33	ИП1	61
02	ИП8	68	18	С/П	50	34	—	11
03	+	10	19	ИП6	66	35	ИП2	62
04	П8	48	20	ИП4	64	36	÷	13
05	ИП1	61	21	×	12	37	П2	42
06	ИП5	65	22	ИП5	65	38	ИП4	64
07	+	10	23	FX ²	22	39	×	12
08	П5	45	24	—	11	40	/—/	0
09	ИП1	61	25	П2	42	41	ИП7	67
10	FX ²	22	26	ИП5	65	42	+	10
11	ИП6	66	27	ИП8	68	43	ИП5	65
12	+	10	28	×	12	44	÷	13
13	П6	46	29	П1	41	45	П3	43
14	ИП2	62	30	ИП6	66	46	С/П	50
15	ИП7	67	31	ИП7	67			

Распределение регистров памяти:

Номер регистра	1	2	3	4	5	6	7	8
Перемен-ная	x_i	y_i, c_0, c_1	c_1	$m+1$	$\sum_{i=0}^m x_i$	$\sum_{i=0}^m x_i^2$	$\sum_{i=0}^m y_i$	$\sum_{i=0}^m x_i y_i$

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) занести в память число узлов аппроксимации и очистить регистры 5—8:

($m + 1$) П4 0 П5 П6 П7 П8

3. Ввести данные для очередного узла аппроксимации по схеме

(x_0) П1 (y_0) В/О С/П

4. Повторить п. 3 для всех узлов ($m + 1$) раз.

5. После последнего ОСТАНОВа нажать клавишу С/П и после ОСТАНОВа записать с индикатора значение c_1 , а из Р2 значение c_0 : ИП2 (c_0).

Контрольный пример. Пусть функция $f(x)$

x_i	0,1	0,2	0,3	0,4	0,5	0,6
y_i	3,7	4,1	4,4	4,8	5,3	5,7

Требуется найти аппроксимирующую функцию $\varphi(x) = c_0 + c_1x$ методом наименьших квадратов. Число узлов $m + 1 = 6$.

После выполнения п. 1, 2 вводим данные для узлов:

0,1 П1 3,7 В/О С/П, после ОСТАНОВа:

0,2 П1 4,1 В/О С/П, после ОСТАНОВа:

0,3 П1 4,4 В/О С/П, после ОСТАНОВа:

0,4 П1 4,8 В/О С/П, после ОСТАНОВа:

0,5 П1 5,3 В/О С/П, после ОСТАНОВа:

0,6 П1 5,7 В/О С/П, после ОСТАНОВа: С/П.

Прочесть на индикаторе значение $c_1 = 4$.

Нажав клавиши ИП2, прочесть на индикаторе значение $c_0 = 3,267$.

Ответ: $\varphi(x) = 3,267 + 4x$. Время счета с вводом исходных данных — около 1,5 мин.

Программа 19 вычисления коэффициентов квадратичного аппроксимирующего многочлена

$$\varphi(x) = c_0 + c_1x + c_2x^2$$

методом наименьших квадратов в автоматическом режиме. В предложенной программе вначале, вычисляются коэффициенты расширенной матрицы системы (5.41) (команды 00—19), а затем система алгебраических уравнений решается методом Гаусса—Жордана.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Командда	Код
00	С/П	50	31	48	48	62	ХУ	14
01	ПА	4—	32	ИП6	66	63	ПС	4—
02	1	01	33	ИП7	67	64	ИПД	6—
03	ПВ	4—	34	ИП8	68	65	П7	47
04	9	09	35	ПП	53	66	В/0	52
05	П1	41	36	59	59	67	ИПС	6—
06	П2	42	37	ПП	53	68	÷	13
07	ПП	53	38	48	48	69	ПД	48
08	79	79	39	ИП5	65	70	ИП8	6—
09	ПП	53	40	ИПА	6—	71	×	12
10	79	79	41	ИПВ	6—	72	—	11
11	С/П	50	42	ПП	53	73	ХУ	14
12	ПВ	4—	43	59	59	74	ИПД	6—
13	ИП9	69	44	ПП	53	75	ИП7	67
14	+	10	45	48	48	76	×	12
15	П9	49	46	ИП9	69	77	—	11
16	ПП	53	47	С/П	50	78	В/0	52
17	79	79	48	ИП3	63	79	ПП	53
18	FLO	5—	49	ИП4	64	80	84	84
19	00	00	50	ИП9	69	81	ПП	53
20	ИП5	65	51	ПП	53	82	84	84
21	ИП6	66	52	67	67	83	В/О	52
22	ИП7	67	53	П4	44	84	ИПА	6—
23	ПП	53	54	ХУ	14	85	ИПВ	6—
24	67	67	55	П9	49	86	×	12
25	ПА	4—	56	ИПД	6—	87	ПВ	4—
26	ХУ	14	57	П3	43	88	КИП1	Г1
27	ПВ	4—	58	В/0	52	89	+	10
28	ИПД	6—	59	ПП	53	90	КП2	—2
29	П5	45	60	67	67	91	В/0	52
30	ПП	53	61	П8	48			

Распределение регистров памяти:

Номер регистра	3	4	9	5	6	7	8
Перемен-ная	$\sum_{i=0}^m x_i^2 y_i, c_2$	$\sum_{i=0}^m x_i y_i, c_1$	$\sum_{i=0}^m y_i, c_0$	$\sum_{i=0}^m x_i^4$	$\sum_{i=0}^m x_i^3$	$\sum_{i=0}^m x_i^2$	$\sum_{i=0}^m x_i$

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) очистить регистры памяти 0 П3 П4 П5 П6 П7 П8 П9 и ввести число узлов аппроксимации

$(m + 1)$ ПС ПО

3. Нажать клавиши В/О С/П. При каждом ОСТАНОВЕ ввести последовательно данные для узлов аппроксимации:

(x_0) С/П (y_0) С/П ... (x_m) С/П (y_m) С/П

4. После последнего ОСТАНОВа прочесть на индикаторе или в Р9 значение c_0 , нажав клавиши ИП4 — значение c_1 , клавиши ИП3 — значение c_2 .

Ответ: $\varphi(x) = c_0 + c_1x + c_2x^2$.

Контрольный пример. Пусть функция $f(x)$

x_i	-2	-1	0	1	2
y_i	4	2	3	6	8

$m + 1 = 5$ — число узлов.

1. Нажимаем клавиши F ПРГ, вводим программу.

2. Нажимаем клавиши F АВТ и клавиши

0 П3 П4 П5 П6 П7 П8 П9 5 ПС ПО

3. Нажимаем клавиши В/О С/П

После ОСТАНОВа последовательно вводим данные для узлов:

2 /—/ С/П 4 С/П 1 /—/ С/П 2 С/П 0 С/П 3 С/П

1 С/П 6 С/П 2 С/П 8 С/П

4. Запишем с индикатора $c_0 = 3,1714286$,

ИП4 — запишем $c_1 = 1,2$;

ИП3 — запишем $c_2 = 0,7142857$.

Ответ: $\varphi(x) = 3,171 + 1,200x + 0,714x^2$. Время вычислений с вводом исходных данных — около 2 мин.

Задания к лабораторной работе 6. Методом наименьших квадратов найти:

а) коэффициенты линейной регрессии

$$\varphi(x) = c_0 + c_1x$$

для функции

x_i	0	0,1	0,2	0,3
y_i	$0,01 + 0,1(k-1)$	$0,19 + 0,1(k-1)$	$0,31 + 0,1(k-1)$	$0,42 + 0,1(k-1)$
	0,4			
	$0,50 + 0,1(k-1)$			

где $k = 1, 2, 3, \dots, 30$ — номер варианта;

б) коэффициенты квадратичного аппроксимирующего многочлена

$$\varphi(x) = c_0 + c_1x + c_2x^2$$

для функции

x_i	2	4	6	8
y_i	-0,24 $\pm k$	0,44 $\pm k$	1,04 $\pm k$	1,56 $\pm k$
	-0,24 $\pm 0,1l$	0,44 $\pm 0,1l$	1,04 $\pm 0,1l$	1,56 $\pm 0,1l$
	-2,14 $\pm 0,2r$	-1,46 $\pm 0,2r$	-0,86 $\pm 0,2r$	-0,34 $\pm 0,2r$
x_i	10	12	14	
y_i	2,0 $\pm k$	2,36 $\pm k$	2,64 $\pm k$	
	2,0 $\pm 0,1l$	2,36 $\pm 0,1l$	2,64 $\pm 0,1l$	
	0,10 $\pm 0,2r$	0,46 $\pm 0,2r$	0,74 $\pm 0,2r$	

где $k = 1, 2, 3, \dots, 10; l = 11, 12, \dots, 19; r = 20, 21, \dots, 30$ — номера вариантов.

5.8. Точечная аппроксимация функций ортогональными полиномами

Пусть функция $f(x)$ задана своими значениями на множестве X точек x_0, x_1, \dots, x_n . Будем искать аппроксимирующую функцию $\varphi(x)$ в виде обобщенного полинома $\varphi(x) = \sum_{i=0}^m c_i \varphi_i(x)$, а базисные функции $\{\varphi_i(x)\}_0^n$ выберем на множестве X так, чтобы при увеличении порядка полинома все предыдущие коэффициенты c_i не изменялись. Этого можно добиться, если выбрать функции $\varphi_i(x)$ ортогональными на X . В частности, в качестве базиса можно выбрать систему полиномов Чебышева последовательно возрастающих степеней, ортогональных с весом на множестве точек X , т. е. положить

$$\varphi_i(x) = P_{im}(x),$$

где индекс $i = 0, 1, \dots, n$ определяет степень многочлена, m — число узлов, уменьшенное на единицу. Условия ортогональности системы полиномов на множестве точек X имеют вид

$$\sum_{k=0}^m p_k P_{im}(x_k) P_{jm}(x_k) = 0 \quad (i \neq j; i, j = 0, 1, \dots, n), \quad (5.44)$$

где $p_k > 0$ — весовые коэффициенты.

При $n \leq m$ множество таких полиномов с ненулевыми нормами

$$\sum_{k=0}^m p_k P_{im}^2(x_k) > 0 \quad (i = 0, 1, \dots, n)$$

образует базис в пространстве H_{n+1} [15].

Аппроксимирующая функция

$$\varphi(x) = c_0 P_{0m}(x) + c_1 P_{1m}(x) + \cdots + c_n P_{nm}(x) \quad (5.45)$$

будет многочленом степени n .

Определим параметры c_i ($i = 0, 1, \dots, n$) методом наименьших квадратов из условия минимума среднего квадратического отклонения. Тогда, согласно (5.28) получим

$$c_i = \frac{\sum_{k=0}^m p_k f(x_k) P_{im}(x_k)}{\sum_{k=0}^m p_k P_{im}^2(x_k)} \quad (i = 0, 1, \dots, n). \quad (5.46)$$

Коэффициенты c_i , определяемые по последним формулам, будут обобщенными коэффициентами Фурье функции $f(x)$ относительно системы ортогональных полиномов $\{P_{im}(x)\}_{i=0}^n$.

В данном случае легко подсчитать, учитывая (5.29), среднее квадратическое отклонение функций $f(x)$ и $\varphi(x)$ на множестве X :

$$\delta_n = \|f - \varphi\| = (\|f\|^2 - \sum_{i=0}^n c_i^2 \|P_{im}\|^2)^{1/2}, \quad (5.47)$$

$$\text{где } \|f\|^2 = \sum_{k=0}^m p_k f^2(x_k); \|P_{im}\|^2 = \sum_{k=0}^m p_k P_{im}^2(x_k).$$

Отсюда следует, что с увеличением n погрешность аппроксимации уменьшается. Поэтому, если при выбранном n точность неудовлетворительна, повышают степень полинома $\varphi(x)$.

Учитывая свойство минимальности коэффициентов Фурье, отметим, что обобщенный полином (5.45) порядка n с коэффициентами Фурье функции $f(x)$, определяемыми по формулам (5.46), имеет наименьшее среднее квадратическое отклонение от $f(x)$ по сравнению с любыми другими полиномами того же порядка n [24].

Ортогональные полиномы Чебышева $P_{im}(x)$ ($i = 0, 1, \dots, n$), зависящие от выбора узлов x_k и весов p_k ($k = 0, 1, \dots, m$), определяются с точностью до постоянного множителя. Если, например, потребовать, чтобы коэффициент при наивысшей степени x равнялся единице, то они определяются однозначно. Такие полиномы при любом расположении узлов можно построить из последовательности $1, x, x^2, \dots$ с помощью ортогонализации Шмидта [20].

Если веса $p_k = 1$ ($k = 0, 1, \dots, m$), а точки x_0, x_1, \dots, x_m — равноотстоящие с шагом h , то для полиномов Чебышева и их значений составлены таблицы [12]. С помощью подстановки

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

точки $x = x_0, x_1, \dots, x_m$ переводятся в точки $t = 0, 1, 2, \dots, m$. На множестве таких целочисленных узлов полиномы Чебышева и их нормы вычисляются по таким формулам [8, 15]:

$$P_{im}(t) = \sum_{k=0}^l (-1)^k C_i^k C_{i+k}^k \frac{t^{[k]}}{m^{[k]}}, \quad (5.48)$$

$$\|P_{im}(t)\|^2 = \sum_{k=0}^m P_{im}^2(k) = \frac{(m+i+1)^{[l+1]}}{(2i+1)m^{[i]}}, \quad (5.49)$$

$$i = 0, 1, 2, \dots, n,$$

где используются сокращенные обозначения для факториальных полиномов

$$t^{[k]} = t(t-1)\dots(t-k+1),$$

C_q^k — число комбинаций из q по k .

По формулам (5.48) выпишем первые пять полиномов Чебышева:

$$P_{0m}(t) = 1; \quad P_{1m}(t) = 1 - 2 \frac{t}{m}; \quad P_{2m}(t) = 1 - 6 \frac{t}{m} + \\ + 6 \frac{t(t-1)}{m(m-1)};$$

$$P_{3m}(t) = 1 - 12 \frac{t}{m} + 30 \frac{t(t-1)}{m(m-1)} - 20 \frac{t(t-1)(t-2)}{m(m-1)(m-2)};$$

$$P_{4m}(t) = 1 - 20 \frac{t}{m} + 90 \frac{t(t-1)}{m(m-1)} - 140 \frac{t(t-1)(t-2)}{m(m-1)(m-2)} + \\ + 70 \frac{t(t-1)(t-2)(t-3)}{m(m-1)(m-2)(m-3)}.$$

В окончательном результате возвращаемся к переменной x по формуле

$$t = \frac{x - x_0}{h}.$$

Пример. С помощью полиномов Чебышева аппроксимировать функцию (см. контрольный пример программы 19):

y_i	4	2	3	6	8
x_i	-2	-1	0	1	2
t_i	0	1	2	3	4

С помощью подстановки $t = x + 2$ получим целочисленные узлы $t = 0, 1, 2, 3, 4$. Выберем $n = 2$, т. е. используем первые три полинома Чебышева $P_{04}(t) = 1; P_{14}(t) = 1 - \frac{t}{2}; P_{24}(t) = 1 - \frac{3}{2}t + \frac{1}{2}t(t-1)$.

Вычисления будем проводить на микрокалькуляторе в непрограммируемом режиме. Учитывая (5.49), вычислим квадраты норм полиномов

$$\|P_{04}\|^2 = 5; \|P_{14}\|^2 = \frac{6 \cdot 5}{3 \cdot 4} = 2,5; \|P_{24}\|^2 = \frac{7 \cdot 6 \cdot 5}{5 \cdot 4 \cdot 3} = 3,5.$$

Параметры аппроксимации определяем по (5.46) ($p_k = 1, k = 0, 1, \dots, m$):

$$c_0 = \frac{1}{5} (4 + 2 + 3 + 6 + 8) = 4,6;$$

$$c_1 = \frac{1}{2,5} (4 \cdot 1 + 2 \cdot 0,5 + 3 \cdot 0 + 6 \cdot (-0,5) + 8 \cdot (-1)) = -2,4;$$

$$c_2 = \frac{1}{3,5} (4 \cdot 1 + 2 \cdot (-0,5) + 3 \cdot (-1) + 6 \cdot (-0,5) + 8 \cdot 1) = 1,4286.$$

Аппроксимирующий полином

$$\varphi(t) = 4,6 - 2,4 \left(1 - \frac{1}{2}t\right) + 1,4286 \left(1 - \frac{3}{2}t + \frac{1}{2}t(t-1)\right).$$

Положив $t = x + 2$, выражим его через x :

$$\varphi(x) = 3,171 + 1,200x + 0,714x^2.$$

Аналогичный результат был получен при решении нормальной системы уравнений. Учитывая (5.47), вычислим квадрат отклонений полинома $\varphi(x)$ от табличных данных:

$$\delta_2^2 = (16 + 4 + 9 + 36 + 64) - (4,6^2 \cdot 5 + 2,4^2 \cdot 2,5 + 1,4286^2 \cdot 3,5) = 129 - 127,343 = 1,657.$$

Поскольку отклонения $\varphi(x)$ от табличных данных довольно значительны, повысим степень полинома $\varphi(x)$ на единицу. Добавим полином

$$P_{34}(t) = 1 - 3t + \frac{5}{2}t(t-1) - \frac{5}{6}t(t-1)(t-2).$$

Вычислим квадрат его нормы

$$\|P_{34}\|^2 = \frac{8 \cdot 7 \cdot 6 \cdot 5}{7 \cdot 4 \cdot 3 \cdot 2} = 10.$$

При этом требуется вычислить всего один коэффициент

$$c_3 = \frac{1}{10} (4 \cdot 1 + 2(-2) + 3 \cdot 0 + 6 \cdot 2 + 8(-1)) = 0,4.$$

Таким образом, аппроксимирующий кубический полином

$$\begin{aligned}\varphi(t) = & 4,6 - 2,4 \left(1 - \frac{1}{2}t\right) + 1,4286 \left(1 - \frac{3}{2}t + \frac{1}{2}t(t-1)\right) + \\ & + 0,4 \left(1 - 3t + \frac{5}{2}t(t-1) - \frac{5}{6}t(t-1)(t-2)\right)\end{aligned}$$

или через переменную x

$$\varphi(x) = 3,171 + 2,333x + 0,714x^2 - 0,333x^3.$$

Квадрат отклонений кубического полинома от табличных данных

$$\delta_3^2 = 0,057.$$

Замечание. Погрешность аппроксимации для случая дискретного множества точек x_0, x_1, \dots, x_m состоит из погрешности метода и погрешности результатов наблюдений. Погрешность метода наименьших квадратов вызвана тем, что функция $f(x) \notin H_{n+1}$ заменяется обобщенным полиномом $\varphi(x) \in H_{n+1}$. Точность аппроксимации зависит от самой функции $f(x)$ и от выбора базиса $\{\varphi_i(x)\}_0^n$.

Как было показано выше для системы ортогональных полиномов, квадрат отклонения δ_n^2 функций $\varphi(x)$ и $f(x)$ уменьшается с увеличением порядка полинома n .

Погрешность результатов наблюдений вызвана случайными ошибками в измеренных значениях функции $f(x_i)$. Она также зависит от выбора числа узлов аппроксимации, их расположения и от выбора весовых коэффициентов. Если увеличивать порядок полинома n , то при $n = m$ будет интерполироваться функция $f(x)$, при этом не сглаживается шум эксперимента, т. е. случайные ошибки в исходных данных будут сильно искажать интерполяционный полином. Если же выбрать порядок полинома низким, то получим грубое описание экспериментальных зависимостей. Дисперсия случайной ошибки растет [8] с увеличением порядка полинома и убывает с увеличением числа узлов, поэтому желательно выбирать $n \ll m$.

В математической статистике разработаны специальные критерии для проверки гипотезы о наличии параболической регрессии n -го порядка.

В практических вычислениях порядок полинома можно выбрать равным порядку практически постоянных разностей табличных значений функции. Кроме того, можно последовательно повышать порядок полинома, сравнивая квадраты отклонений δ_n^2 с погрешностью результатов эксперимента ε . При $\delta_n^2 > \varepsilon$ увеличиваем n пока не достигнем $\delta_n^2 \approx \varepsilon$. Если при этом окажется, что $n \ll m$, то аппроксимация проведена удачно. Если n почти совпадает с числом узлов, то следует выбрать другие базисные функции.

5.9. Точечная аппроксимация функции тригонометрическими полиномами

Во многих технических задачах требуется аппроксимировать периодическую функцию $f(x)$ периода $T = 2l$. В таких случаях приближение целесообразно находить в виде тригонометрического полинома порядка n по системе ортогональных на отрезке $[0, 2l]$ функций

$$1, \cos \frac{\pi x}{l}, \sin \frac{\pi x}{l}, \cos \frac{2\pi x}{l}, \sin \frac{2\pi x}{l}, \dots, \cos \frac{n\pi x}{l}, \sin \frac{n\pi x}{l}, \dots \quad (5.51)$$

При некоторых ограничениях на функцию $f(x)$ ее можно представить в виде суммы тригонометрического ряда Фурье

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos \frac{k\pi x}{l} + b_k \sin \frac{k\pi x}{l} \right). \quad (5.52)$$

Коэффициенты этого ряда вычисляются по формулам Эйлера—Фурье:

$$\begin{cases} a_0 = \frac{1}{l} \int_0^{2l} f(x) dx; \\ a_k = \frac{1}{l} \int_0^{2l} f(x) \cos \frac{k\pi x}{l} dx, \quad k = 1, 2, \dots; \\ b_k = \frac{1}{l} \int_0^{2l} f(x) \sin \frac{k\pi x}{l} dx, \quad k = 1, 2, \dots. \end{cases} \quad (5.53)$$

Сформулируем достаточные условия сходимости ряда Фурье (5.52) к функции $f(x)$.

Теорема Дирихле. Если функция $f(x)$ периодична с периодом $2l$ и на интервале $[0, 2l]$ кусочно-монотонна и кусочно-непрерывна, то ряд Фурье этой функции сходится на всей числовой оси, и его сумма равна:

$f(x)$ во всех точках непрерывности функции;
полусумме односторонних пределов функции $\frac{1}{2} [f(x - 0) + f(x + 0)]$ во всех точках разрыва.

Если кусочно-монотонная функция $f(x)$ периода $2l$ всюду непрерывна, то ряд Фурье сходится абсолютно и равномерно.

Коэффициенты (5.53) сходящегося ряда Фурье убывают по модулю при увеличении k , причем скорость убывания зависит от степени гладкости функции $f(x)$. Так, если $f(x)$ имеет разрывы первого рода, то (a_k) и (b_k) убывают со скоростью $1/k$. Если $f(x)$ непрерывна, но ее производная $f'(x)$ имеет разрывы первого рода, то скорость убывания модулей коэффициентов равна $1/k^2$.

Разложение функции $f(x)$ в ряд Фурье позволяет представить ее в виде суммы бесконечного числа гармоник

$$a_k \cos \frac{k\pi x}{l} + b_k \sin \frac{k\pi x}{l} = A_k \sin \left(\frac{k\pi x}{l} + \varphi_k \right) \quad (k = 1, 2, \dots).$$

Их амплитуды A_k и начальные фазы φ_k выражаются через коэффициенты ряда:

$$A_k = \sqrt{a_k^2 + b_k^2}, \quad \operatorname{tg} \varphi_k = \frac{a_k}{b_k}.$$

Первая гармоника ($k = 1$) — основная, в частности, в акустике ее называют основным тоном. Остальные гармоники определяют обертоны, с увеличением k их влияние уменьшается (они создают тембр звука).

Представление функции $f(x)$ рядом Фурье называется гармоническим анализом. В практических вычислениях бесконечный ряд Фурье (5.52) заменяют приближением его n -й частичной суммой, т. е. функцию $f(x)$ аппроксимируют тригонометрическим полиномом n -го порядка

$$\varphi(x) \equiv S_n(x) = \frac{a_0}{2} + \sum_{k=1}^n \left(a_k \cos \frac{k\pi x}{l} + b_k \sin \frac{k\pi x}{l} \right). \quad (5.54)$$

Вычисление коэффициентов Эйлера — Фурье (5.53) возможно в принципе для аналитически заданной функции, однако выполнить интегрирование в конечном виде удается редко. Практически в гармоническом анализе их вычисляют

приближенно по какой-либо формуле численного интегрирования (см. гл. 6).

В большинстве инженерных задач функция $f(x)$ задается, как правило, на промежутке длины периода $[0; 2l]$ в виде таблицы или графика. Пусть $f(x)$ задана своими значениями $f(x_i) = y_i$ ($i = 0, 1, 2, \dots, m$) на множестве X равноотстоящих точек отрезка $[0; 2l]$:

$$x_0 = 0, x_1 = \frac{2l}{m+1}, \dots, x_m = \frac{m \cdot 2l}{m+1}. \quad (5.55)$$

Рассмотрим задачу среднего квадратического приближения функции тригонометрическим полиномом (5.54) порядка n по методу наименьших квадратов. Полагаем количество неизвестных параметров полинома $\varphi(x)$, равное $2n + 1$, не превышающим числа узлов $m + 1$, т. е. $2n \leq m$. Как было указано выше, вычисление коэффициентов полинома $\varphi(x)$ (5.54) существенно упрощается, если система базисных функций ортогональна на заданном множестве точек.

Если выбрать веса $p_i = 1$ ($i = 0, 1, \dots, m$) и узлы равноотстоящими, то при $2n \leq m$ система базисных функций (5.51) будет ортогональной на дискретном множестве точек (5.55) при любом m [4].

Действительно, воспользуемся формулой Эйлера $\cos \varphi + i \sin \varphi = e^{i\varphi}$ и вытекающим отсюда тождеством $e^{2k\pi i} = 1$.

Покажем, что $\left(1, \cos \frac{k\pi x}{l}\right) = 0$ и $\left(1, \sin \frac{k\pi x}{l}\right) = 0$. По определению скалярного произведения функций на дискретном множестве точек имеем:

$$\begin{aligned} \left(1, \cos \frac{k\pi x}{l}\right) + i \left(1, \sin \frac{k\pi x}{l}\right) &= \sum_{j=0}^m \cos \frac{2k\pi jl}{l(m+1)} + \\ &+ i \sum_{j=0}^m \sin \frac{2k\pi jl}{l(m+1)} = \sum_{j=1}^m e^{i \cdot \frac{2k\pi j}{m+1}} = \frac{1 - e^{i \cdot \frac{2k\pi(m+1)}{m+1}}}{1 - e^{i \cdot \frac{2k\pi}{m+1}}} = \frac{1 - e^{2k\pi i}}{1 - e^{\frac{2k\pi i}{m+1}}} = 0. \end{aligned}$$

Следовательно, действительная и мнимая части в стоящем слева выражении равны нулю. Докажем ортогональность функций $\cos \frac{k\pi x}{l}$ и $\cos \frac{r\pi x}{l}$ при $k \neq r$, $k, r = 1, 2, \dots, n$:

$$\left(\cos \frac{k\pi x}{l}, \cos \frac{r\pi x}{l}\right) = \sum_{j=0}^m \cos \frac{2k\pi lj}{l(m+1)} \cdot \cos \frac{2r\pi lj}{l(m+1)} =$$

$$\begin{aligned}
&= \frac{1}{2} \sum_{i=0}^m \cos(k+r) \frac{2\pi j}{m+1} + \frac{1}{2} \sum_{i=0}^m \cos(k-r) \frac{2\pi j}{m+1} = \\
&= \frac{1}{2} \left(1, \cos \frac{(k+r)\pi x}{l} \right) + \frac{1}{2} \left(1, \cos \frac{(k-r)\pi x}{l} \right) = 0 \text{ при } k \neq r.
\end{aligned}$$

Аналогично доказываются равенства

$$\left(\cos \frac{k\pi x}{l}, \sin \frac{r\pi x}{l} \right) = 0, \quad \left(\sin \frac{k\pi x}{l}, \sin \frac{r\pi x}{l} \right) = 0$$

при $k \neq r$ ($k, r = 1, 2, \dots, n$).

Вычислим квадраты норм базисных функций

$$\begin{aligned}
\|1\|^2 &= \sum_{j=0}^m 1 = m+1; \quad \left\| \cos \frac{k\pi x}{l} \right\|^2 = \sum_{j=0}^m \cos^2 \frac{k\pi j + 2l}{l(m+1)} = \\
&= \frac{1}{2} \sum_{j=0}^m \left(1 + \cos \frac{4k\pi j}{m+1} \right) = \frac{m+1}{2} \quad (k = 1, 2, \dots, n);
\end{aligned}$$

аналогично $\left\| \sin \frac{k\pi x}{l} \right\|^2 = \frac{m+1}{2}$ ($k = 1, 2, \dots, n$); $2n \leq m$.

Учитывая формулы (5.28) для вычисления обобщенных коэффициентов Фурье по ортогональной системе функций, определяем параметры тригонометрического полинома $\varphi(x)$:

$$\left\{
\begin{array}{l}
a_0 = \frac{2}{m+1} \sum_{i=0}^m f(x_i); \\
a_k = \frac{2}{m+1} \sum_{i=0}^m f(x_i) \cos \frac{2k\pi i}{m+1}, \quad k = 1, 2, \dots, n; \\
b_k = \frac{2}{m+1} \sum_{i=0}^m f(x_i) \sin \frac{2k\pi i}{m+1}, \quad k = 1, 2, \dots, n.
\end{array} \right. \quad (5.56)$$

Равенства (5.56) называют формулами Бесселя. Они могут быть получены из (5.53) для коэффициентов ряда Фурье функции $f(x)$ периода $2l$, если к вычислениям интегралов применить формулу трапеций (см. п. 6.3) с узлами $x_i = \frac{2li}{m+1}$ ($i = 0, 1, \dots, m$) и считать $f(0) = f(2l)$.

Если число узлов аппроксимации равно числу неизвестных параметров тригонометрического полинома $\varphi(x)$, т. е. $m+1 = 2n+1$, то $\varphi(x)$ будет интерполяционным тригонометрическим полиномом порядка n для функции $f(x)$. Такой

полином однозначно определяется значениями функции в узлах x_i ($i = 0, 1, \dots, m$) промежутка $[0; 2l]$. Как и в случае квадратичной полиномиальной аппроксимации, для сглаживания экспериментальных данных $f(x_i)$ ($i = 0, 1, \dots, m$), вычисленных со случайными ошибками, целесообразно выбирать порядок n полинома значительно меньшим числа узлов, т. е. $2n \ll m$.

Тригонометрический полином $\varphi(x)$, коэффициенты которого вычислены по формулам Бесселя (5.56), будет наилучшим средним квадратическим приближением функции $f(x)$ среди всех тригонометрических полиномов n -го порядка, т. е. он обеспечивает минимум суммы квадратов отклонений

$$\delta_n^2 = \sum_{i=0}^m [\varphi(x_i) - f(x_i)]^2.$$

При увеличении порядка n полинома $\varphi(x)$ δ_n^2 убывает, но следует учитывать то, что при этом возрастает погрешность за счет влияния случайных ошибок в измеренных значениях функции, а также вычислительная погрешность. Поэтому для выбора оптимального порядка n тригонометрического аппроксимирующего полинома можно воспользоваться следующим приемом [8]. Если в формуле (5.54) последовательно прибавлять k -е гармоники и вычислять соответствующие им амплитуды A_k , то до некоторого значения k амплитуды резко убывают, а затем остаются почти неизменными. Таким значением k ограничиваем порядок полинома $\varphi(x)$.

Для функции $f(x)$ периода $T = 2\pi$ можно выбрать число узлов кратным четырем. Тогда вычисление коэффициентов аппроксимации можно упростить, так как значения $\sin kx_i$ и $\cos kx_i$ будут повторяться.

Если функция $f(x)$ не является периодической, а задана на некотором конечном интервале, то с помощью параллельного переноса его можно перевести в отрезок $[0; l]$. Считая этот отрезок полупериодом функции, доопределим ее на отрезке $[-l; 0]$ четным способом, полагая $f(-x) = f(x)$, а затем периодически продолжим с периодом $T = 2l$ на всю числовую ось. К такой функции можно применять изложенную выше методику аппроксимации тригонометрическим полиномом по известным ее значениям на множестве равноотстоящих точек отрезка $[0; l]$.

Реализация вычислений коэффициентов аппроксимирующего тригонометрического полинома по формулам Бесселя (5.66) осуществляется программами 20 и 21, а приближенное вычисление значений аппроксимирующего тригонометрического полинома $\varphi(x)$ в любой точке $x \in [0; T]$ программой 22.

5.10. Лабораторная работа 7. Аппроксимация функции тригонометрическими полиномами по методу наименьших квадратов

Программа 20 вычисления коэффициентов аппроксимирующего тригонометрического полинома по формулам (5.56) для периодической функции $f(x)$, с периодом T . Рассматривается $m + 1$ точка интервала $[0; T]$.

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИПВ	6L	22	ИП	53	44	+-	10
01	ИП7	67	23	29	29	45	ПА	4-
02	÷	13	24	БП	51	46	ИПС	6-
03	ПВ	4L	25	14	14	47	ИПВ	6L
04	Fп	20	26	Фп	20	48	+-	10
05	2	02	27	2	02	49	ПС	4-
06	×	12	28	÷	13	50	ИП8	68
07	ИП7	67	29	Г16	46	51	ИП9	69
08	÷	13	30	0	00	52	+	10
09	ПД	4Г	31	П8	48	53	П8	48
10	0	00	32	ПС	4-	54	FЛ0	5-
11	П9	49	33	ПА	4-	55	36	36
12	ПП	53	31	ИП7	67	56	ИПА	6-
13	26	26	35	П0	40	57	2	02
14	ИП9	69	36	П11	53	58	×	12
15	ИПД	6Г	37	63	63	59	ИП7	67
16	+	10	38	ИП18	68	60	÷	13
17	П9	49	39	ИП6	66	61	С/П	50
18	0	00	40	+	10	62	В/О	52
19	ПП	53	41	F sin	1-	63	... } f(x)	...
20	26	26	42	×	12 } f(x)	...
21	0	60	43	ИПА	6-	...	В/О	...

Распределение регистров памяти:

Номер регистра	A	B	C	D	q	8	7	1-5
Переменная	Σ	$T, \frac{T}{m+1}$	x	$\frac{2\pi}{m+1}$	$\frac{2\pi k}{m+1}$	$\frac{2\pi ki}{m+1}$	$m+1$	свободны

Инструкция. 1. Переключатель Р—Г поставить в положение Р.

2. Написать подпрограмму вычисления значений $f(x)$. Значение x хранится в РС.

3. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

4. В автоматическом режиме (F АВТ) ввести данные:

(T) ПВ ($m + 1$) П7

5. Приступить к вычислениям:

нажать клавиши В/О С/П, после останова списать a_0 ;

нажать клавишу С/П, после останова списать a_1 ;

нажимая клавишу С/П, после каждого останова списать поочередно b_1, a_2, b_2, a_3 и т. д.

Контрольный пример. Вычислить коэффициенты аппроксимирующего тригонометрического полинома третьего порядка для функции $y = e^{\cos x}$. Рассмотреть $m + 1 = 8$ точек. Отметим, что данная функция периодична с периодом 2π .

Составим подпрограмму вычисления значений функции:

ИПС Fcos Fe^x В/О

После ввода программы вводим данные:

F π 2 × ПВ 8 П7

и приступаем к вычислениям:

В/О С/П, после останова списываем

$a_0 = 2,5321322$, далее С/П $a_1 = 1.1303216$, С/П

$b_1 = -2.5 \cdot 10^{-8}$, С/П $a_2 = 2.7154005 \cdot 10^{-1}$, С/П

$b_2 = -7.5 \cdot 10^{-8}$, С/П $a_3 = 4.488005 \cdot 10^{-2}$, С/П

$b_3 = -6.25 \cdot 10^{-7}$

Время вычислений — около 9 мин.

По программе 20 значения $f(x_i)$ ($i = 0, 1, \dots, m$) вычисляются заново для каждого из коэффициентов a_k, b_k . Если число узлов выбрать таким, что $m + 1 < 10$, то можно проводить вычисления коэффициентов полинома порядка n (где $2n + 1 < m + 1 < 10$) по более экономной программе 21.

Программа 21 вычисления коэффициентов аппроксимирующего тригонометрического полинома порядка n по формулам Бесселя (5.56) для периодической функции с периодом 2π , если число узлов $m + 1 < 10$ и $2n + 1 < m + 1$. Заданы значения функции в узлах.

ИПД ПО	КИПО	ИПА	КИП↑	+	FL0 04	ПП	50
С/П Fπ	ПС	ИПД ПО	КИПО 0	ИПА ИПО	ИПС		
× ПП	50	ПВ	F cos	КИП↑ ×	+	XУ	ИПВ
Fsin КИП↑ ×	+ XУ		FL0 18	ПП	50	C/П	
XУ ПП	50	С/П	ИПС	Fπ	+	ПС	БП 13
2 × ИПД	÷	B/O					

Распределение регистров памяти:

Номер регистра	D	0	C	B	A	1	...	m
Переменная	$m + 1$	k	πi	$\frac{2\pi ik}{m + 1}$	y_0	y_1	...	y_m

Инструкция. 1. Переключатель Р—Г поставить в положение Р.

2. В режиме программирования (F ПРГ) ввести программу.

3. В автоматическом режиме (F АВТ) ввести данные: $(m + 1)$ ПД (y_0) ПА (y_1) П1 ... (y_m) П (m) .

4. Приступить к вычислениям:

нажать клавиши В/О С/П, после останова списать a_0 ;

нажать клавишу С/П, после останова списать a_1 ;

нажимая клавишу С/П, после каждого останова списать поочередно $b_1, a_2, b_2, \dots, a_n, b_n$.

Контрольный пример и результаты вычислений такие же, как и в программе 20. Время вычислений — около 3 мин.

Программа 22 вычисления значения аппроксимирующего тригонометрического полинома $\varphi(x)$ третьего порядка по формуле (5.54) для функции $f(x)$ периода T , заданной таблично в восьми равноотстоящих точках интервала $[0; T]$:

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	Fп	20	17	—	11	34	1	01
01	4	04	18	Fп	20	35	+	10
02	÷	13	19	×	12	36	8	08
03	ПВ	4L	20	ПА	4—	37	÷	13
04	8	08	21	F cos	1Γ	38	КИП↑	ΓЕ
05	П0	40	22	ИПА	6—	39	×	12
06	0	00	13	↑	OE	40	ИПС	6C
07	ПС	4C	24	+	10	41	+	10
08	ИПА	6—	25	F cos	1Γ	42	ПС	4C
09	2	02	26	BХ	0	43	ИПА	6—
10	×	12	27	ИПА	6—	44	ИПВ	6L
11	ИП9	69	28	+	10	45	+	10
12	÷	13	29	F cos	1Γ	46	ПА	4—
13	1	01	30	+	10	47	FL0	5Γ
14	,	0—	31	+	10	48	2I	2I
15	7	07	32	↑	0E	49	ИПС	6C
16	5	05	33	+	10	50	С/П	50

Распределение регистров памяти:

Номер регистра	0	C	B	A	9	1...8
Переменная	цикл по i	Σ	$\pi/4$	$x, \frac{i\pi}{4}$	T	$y_0 \dots y_7$

Инструкция. 1. Переключатель Р—Г поставить в положение Р.

2. Ввести программу в режиме программирования (F ПРГ).

3. В автоматическом режиме (F АВТ) ввести данные: (y_0) П1 (y_1) П2 ... (y_7) П8 (T) П9 (x) ПА.

4. Нажать клавиши В/О С/П. После останова $\varphi(x)$ прочесть на индикаторе.

Контрольный пример. Периодическая функция $y = f(x)$ задана на отрезке $[0; 4]$ таблицей:

x_i	0	0,5	1	1,5	2	2,5	3	3,5
y_i	2	2,7	2	1,6	1,3	1	1,2	1,4

Найти значение аппроксимирующего тригонометрического полинома третьего порядка в точке $x = 1,2$.

После ввода программы вводим данные:

2 П1 П3 2.7 П2 1.6 П4 1.3 П5 1 П6 1.2 П7 ПА
1.4 П8 4 П9

После нажатия клавиш В/О С/П примерно через 1,5 мин. прочтем на индикаторе 1,747731.

Ответ: $\varphi(1,2) = 1,75$.

Задания к лабораторной работе 7.

1. Вычислить коэффициенты a_0, a_k, b_k ($k = 1, 2, 3$) аппроксимирующего полинома

$$\varphi_3(x) = \frac{a_0}{2} + \sum_{k=1}^3 (a_k \cos kx + b_k \sin kx)$$

для функции

$$f(x) = e^{A \cos(x+x_0)},$$

где $A = \frac{29+n}{30}$, $x_0 = \frac{n}{5}$, n — номер варианта. Рассмотреть $m+1=8$ точек.

Вычисление коэффициентов по формулам Бесселя (5.56) для аналитически заданной функции реализуется программой 20.

2. Функция $y = f(x)$ задана таблично на отрезке $[0; 2\pi]$:

x_i	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$
y_i	0,4	0,6	0,8	0,9	1,0	1,0	0,9	0,2

Найти значение аппроксимирующего тригонометрического полинома третьего порядка в точке $x = \frac{n}{2}$, где n — номер варианта.

5.11. Интегральная аппроксимация функции методом наименьших квадратов

Пусть функция $f(x) \in L_2^p[a; b]$ задана аналитическим выражением на отрезке $[a; b]$. Требуется аппроксимировать ее обобщенным полиномом

$$\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_n\varphi_n(x)$$

по некоторой системе базисных функций $\{\varphi_i(x)\}_{i=0}^n \in L_2^p[a; b]$.

Необходимость решения такой задачи возникает, например, в тех случаях, когда функция $f(x)$ имеет сложный аналитический вид, неудобный для дальнейших приложений. Тогда можно применить интегральную аппроксимацию методом наименьших квадратов.

В соответствии с этим методом коэффициенты c_i ($i = 0, 1, \dots, n$) аппроксимирующего полинома $\varphi(x)$ определяются из условия минимума среднего квадратического отклонения

$$\Phi(c_0, c_1, \dots, c_n) = \int_a^b p(x) [f(x) - \sum_{i=0}^n c_i \varphi_i(x)]^2 dx. \quad (5.57)$$

Коэффициенты c_i ($i = 0, 1, \dots, n$) определяются из системы линейных уравнений (5.26). Следует учесть, что в случае интегральной аппроксимации

$$\begin{aligned} (\varphi_k, \varphi_j) &= \int_a^b p(x) \varphi_k(x) \varphi_j(x) dx, \\ (f, \varphi_k) &= \int_a^b p(x) f(x) \varphi_k(x) dx. \end{aligned} \quad (5.58)$$

Определитель этой системы есть определитель Грама. Если система функций $\varphi_i(x)$ ($i = 0, 1, \dots, n$) линейно независима на $[a; b]$, то составленный для нее определитель Грама отличен от нуля и система имеет единственное решение.

Интегральные приближения алгебраическими многочленами имеют наиболее простой вид. Однако определенные трудности могут возникнуть при вычислении интегралов вида $(x^k, f) = \int_a^b p(x) x^k f(x) dx$, стоящих в правой части системы (5.26).

Как было отмечено выше, матрица системы (5.26) становится диагональной, если в качестве базисных функций $\{\varphi_i(x)\}_{i=0}^n$ в подпространстве $H_{n+1} \subset L_2^p[a; b]$ выбрать систему

попарно ортогональных на $[a; b]$ функций, удовлетворяющих условиям

$$\int_a^b p(x) \varphi_k(x) \varphi_j(x) dx = 0, \quad k \neq j.$$

Тогда параметры аппроксимирующей функции $\varphi(x)$ будут обобщенными коэффициентами Фурье функции $f(x)$ по данной ортогональной системе:

$$c_i = \frac{\int_a^b p(x) \varphi_i(x) f(x) dx}{\int_a^b p(x) \varphi_i^2(x) dx}. \quad (5.59)$$

В качестве такого ортогонального базиса можно выбрать систему алгебраических полиномов, ортогональных с весом на $[a; b]$. Выбор веса $p(x)$ и отрезка $[a; b]$ позволяет получить различные системы ортогональных полиномов.

Классическими примерами таких ортогональных систем являются полиномы Якоби, Чебышева, Лежандра, Лагерра, Эрмита [4]. Приведем их краткую характеристику.

1. Полиномы Якоби $P_n^{(\alpha, \beta)}(x)$ ортогональны на отрезке $[-1; 1]$ с весом $p(x) = (1-x)^\alpha (1+x)^\beta (\alpha, \beta > -1)$. Их можно вычислять по формуле Родрига

$$P_n^{(\alpha, \beta)}(x) = \frac{(-1)^n}{n! 2^n} (1-x)^{-\alpha} (1+x)^{-\beta} \frac{d^n}{dx^n} [(1-x)^{\alpha+n} (1+x)^{\beta+n}] \quad (5.60)$$

или по рекуррентной формуле

$$\begin{aligned} & 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta) P_{n+1}^{(\alpha, \beta)}(x) = \\ & = (2n+\alpha+\beta+1) [(2n+\alpha+\beta)(2n+\alpha+\beta+2)x + \\ & + \alpha^2 - \beta^2] P_n^{(\alpha, \beta)}(x) - 2(n+\alpha)(n+\beta)(2n+\alpha+\beta+ \\ & + 2) P_{n-1}^{(\alpha, \beta)}(x), \quad n = 1, 2, \dots. \end{aligned} \quad (5.61)$$

2. Полиномы Лежандра $P_n(x)$ (частный случай полиномов Якоби при $\alpha = \beta = 0$) ортогональны на отрезке $[-1; 1]$ с весом $p(x) \equiv 1$.

Формула Родрига

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} [(x^2 - 1)^n] \quad (5.62)$$

и рекуррентная формула

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \\ n=1, 2, \dots; P_0(x) = 1, P_1(x) = x. \quad (5.63)$$

Квадраты их норм известны [4]:

$$\|P_n\|^2 = \int_{-1}^1 P_n^2(x) dx = \frac{2}{2n+1}. \quad (5.64)$$

Выпишем несколько первых полиномов Лежандра

$$P_0(x) = 1; P_4(x) = \frac{1}{8} (35x^4 - 30x^2 + 3); \\ P_1(x) = x; P_5(x) = \frac{1}{8} (63x^5 - 70x^3 + 15x); \quad (5.65) \\ P_2(x) = \frac{1}{2} (3x^2 - 1); P_6(x) = \frac{1}{16} (231x^6 - 315x^4 + 105x^2 - 5); \\ P_3(x) = \frac{1}{2} (5x^3 - 3x); P_7(x) = \frac{1}{16} (429x^7 - 693x^5 + 315x^3 - 35x).$$

3. Полиномы Чебышева первого рода $T_n(x)$ (частный случай полиномов Якоби при $\alpha = \beta = -\frac{1}{2}$) ортогональны на отрезке $[-1; 1]$ с весом $p(x) = \frac{1}{\sqrt{1-x^2}}$.

Их явное выражение

$$T_n(x) = \cos(n \arccos x), n = 0, 1, 2, \dots \quad (5.66)$$

и рекуррентная формула

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), n = 1, 2, \dots. \quad (5.67)$$

Квадраты их норм

$$\|T_n\|^2 = \int_{-1}^1 \frac{T_n^2(x)}{\sqrt{1-x^2}} dx = \begin{cases} \pi, & n = 0; \\ \frac{\pi}{2}, & n \neq 0. \end{cases} \quad (5.68)$$

Полином $T_n(x)$ имеет n действительных различных корней

$$x_i = \cos \frac{2i-1}{2n} \pi \quad (i = 1, 2, \dots, n), \quad (5.69)$$

расположенных на интервале $[-1; 1]$.

Приведем выражения полиномов $T_n(x)$ до 7-й степени

$$T_0(x) = 1; T_4(x) = 8x^4 - 8x^2 + 1; \\ T_1(x) = x; T_5(x) = 16x^5 - 20x^3 + 5x;$$

$$T_2(x) = 2x^2 - 1; \quad T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1; \quad (5.70)$$

$$T_3(x) = 4x^3 - 3x; \quad T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x.$$

4. Полиномы Чебышева второго рода $U_n(x)$ (частный случай полиномов Якоби при $\alpha = \beta = \frac{1}{2}$) ортогональны на отрезке $[-1; 1]$ с весом $p(x) = \sqrt{1-x^2}$.

Их явное выражение

$$U_n(x) = \frac{\sin((n+1)\arccos x)}{\sqrt{1-x^2}} \quad (n = 0, 1, 2, \dots) \quad (5.71)$$

и рекуррентная формула (как и (5.67))

$$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x) \quad (n = 1, 2, \dots). \quad (5.72)$$

Квадраты норм

$$\|U_n\|^2 = \int_{-1}^1 U_n^2(x) \sqrt{1-x^2} dx = \frac{\pi}{2} \quad (n = 0, 1, 2, \dots). \quad (5.73)$$

Выпишем их явные выражения до 5-й степени:

$$\begin{aligned} U_0(x) &= 1; \quad U_1(x) = 2x; \quad U_2(x) = 4x^2 - 1; \\ U_3(x) &= 8x^3 - 4x; \quad U_4(x) = 16x^4 - 12x^2 + 1; \\ U_5(x) &= 32x^5 - 32x^3 + 6x. \end{aligned} \quad (5.74)$$

5. Полиномы Лагерра $L_n^{(\alpha)}(x)$ ортогональны на полуоси $[0; +\infty)$ с весом $p(x) = x^\alpha e^{-x}$, где $\alpha > -1$.

Формула Родрига

$$L_n^{(\alpha)}(x) = (-1)^n x^{-\alpha} e^x \frac{d^n}{dx^n} (x^{\alpha+n} e^{-x}) \quad (5.75)$$

и рекуррентная формула

$$\begin{aligned} L_{n+1}^{(\alpha)}(x) &= (x - \alpha - 2n - 1) L_n^{(\alpha)}(x) - n(\alpha + n) L_{n-1}^{(\alpha)}(x), \\ L_0 &= 1, \quad L_1 = x - 1 \quad (n = 1, 2, \dots). \end{aligned} \quad (5.76)$$

Квадраты их норм равны

$$\|L_n^{(\alpha)}\|^2 = \int_0^\infty x^\alpha e^{-x} [L_n^{(\alpha)}(x)]^2 dx = n! \Gamma(\alpha + n + 1), \quad (5.77)$$

в частности, при $\alpha = 0$ $p(x) = e^{-x}$ и

$$\|L_n^{(0)}\|^2 = (n!)^2. \quad (5.78)$$

Выпишем явные выражения нескольких полиномов Лагерра при $\alpha = 0$:

$$L_0^{(0)}(x) = 1; L_1^{(0)}(x) = x - 1; L_2^{(0)}(x) = x^2 - 4x + 2; \quad (5.79)$$

$$L_3^{(0)} = x^3 - 9x^2 + 18x - 6; L_4^{(0)}(x) = x^4 - 16x^3 + 72x^2 - 96x + 24.$$

6. Полиномы Эрмита $\mathcal{H}_n(x)$ ортогональны на всей числовой оси $(-\infty; +\infty)$ с весом $p(x) = e^{-x^2}$.

Формула Родрига

$$\mathcal{H}_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} [e^{-x^2}] \quad (5.80)$$

и рекуррентная формула

$$\begin{aligned} \mathcal{H}_{n+1}(x) &= 2x\mathcal{H}_n(x) + 2n\mathcal{H}_{n-1}(x), \\ \mathcal{H}_0 &= 1; \mathcal{H}_1 = 2x \quad (n = 1, 2, \dots). \end{aligned} \quad (5.81)$$

Квадраты норм полиномов Эрмита

$$\begin{aligned} \|\mathcal{H}_n\|^2 &= \int_{-\infty}^{+\infty} e^{-x^2} \mathcal{H}_n^2(x) dx = 2^n n! \sqrt{n} \\ &\quad (n = 0, 1, 2, \dots). \end{aligned} \quad (5.82)$$

Выпишем полиномы Эрмита до 5-й степени

$$\begin{aligned} \mathcal{H}_0(x) &= 1; \mathcal{H}_1(x) = 2x; \mathcal{H}_2(x) = 4x^2 - 2; \mathcal{H}_3(x) = 8x^3 - 12x; \\ \mathcal{H}_4(x) &= 16x^4 - 48x^2 + 12; \mathcal{H}_5(x) = 32x^5 - 160x^3 + 120x. \end{aligned} \quad (5.83)$$

Известно [4], что все перечисленные выше системы ортогональных полиномов являются полными в пространстве $L_2^p[a; b]$ при соответствующих весовых функциях $p(x)$. Таким образом, для любой функции $f(x) \in L_2^p[a; b]$ последовательность частичных сумм $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$ обобщенного ряда Фурье по системе классических ортогональных полиномов сходится в среднем к функции $f(x)$.

Выберем в методе наименьших квадратов в качестве базисных функций $\varphi_i(x)$ ($i = 0, 1, 2, \dots, n$) соответствующую систему ортогональных полиномов в зависимости от интервала приближения $[a; b]$ и веса $p(x)$. Коэффициенты аппроксимирующего полинома $\varphi(x) = \sum_{i=0}^n c_i \varphi_i(x)$, выражаемые через интегралы по формуле (5.59), вычисляются приближенно по какой-либо квадратурной формуле (см. гл. 6), например по формуле Симпсона. При этом используются явные аналитические выражения для ортогональных полиномов и значения их норм.

Кроме классических ортогональных полиномов в некоторых прикладных задачах (например, в теории кодирования) используется представление функции $f(x)$ в виде суммы ступенчатых функций. При этом в качестве базиса выбирается система функций Хаара, которые определяются следующим образом:

функция нулевого порядка $\chi_{01}(x) = 1$;

функции первого порядка

$$\chi_{11}(x) = \begin{cases} 1, & x \in J_{11}, \\ -1, & x \in J_{12}, \\ 0, & x \notin J_{11} \cup J_{12}; \end{cases}$$

функции k -го порядка

$$\begin{aligned}\chi_{k1}(x) &= 2^{\frac{k-1}{2}} \begin{cases} 1, & x \in J_{k1}, \\ -1, & x \in J_{k2}, \\ 0, & x \notin J_{k1} \cup J_{k2}; \end{cases} \\ \chi_{k, 2^{k-1}}(x) &= 2^{\frac{k-1}{2}} \begin{cases} 1, & x \in J_{k, 2^{k-1}}, \\ -1, & x \in J_{k, 2^k}, \\ 0, & x \notin J_{k, 2^{k-1}} \cup J_{k, 2^k}; \end{cases} \quad (5.84)\end{aligned}$$

где $J_{kl} = \left[\frac{i-1}{2^k}; \frac{i}{2^k} \right)$ ($i = 1, \dots, 2^k$; $k = 1, 2, \dots$).

Система функций Хаара ортонормирована с весом $p(x) = 1$ в пространстве $L_2[0; 1]$, т. е.

$$(\chi_{ki}, \chi_{lj}) = \int_0^1 \chi_{ki}(x) \chi_{lj}(x) dx = 0 \text{ при } k \neq l \text{ или } i \neq j;$$

$$\|\chi_{ki}\|^2 = \int_0^1 \chi_{ki}^2(x) dx = 1.$$

Эта система является полной в $L_2[0; 1]$ [20]. Поэтому для любой функции $f(x) \in L_2[0; 1]$ ряд

$$\sum_{k=0}^{\infty} \sum_{i=j}^{2^k-1} c_{ki} \chi_{ki}(x), \quad (5.85)$$

где

$$c_{kj} = (f, \chi_{kj}) = \int_0^1 f(x) \chi_{kj}(x) dx, \quad (5.86)$$

сходится в метрике $L_2 [0; 1]$ к $f(x)$, т. е. среднее квадратическое отклонение

$$\delta_n = \left(\int_0^1 \left[f(x) - \sum_{k=0}^n \sum_{j=1}^{2^k-1} c_{kj} \chi_{kj}(x) \right]^2 dx \right)^{1/2}$$

стремится к нулю при $n \rightarrow \infty$. Кроме того, если функция $f(x)$ непрерывна на отрезке $[0; 1]$, то ряд (5.85) сходится к $f(x)$ равномерно, т. е.

$$\left| f(x) - \sum_{k=0}^n \sum_{j=1}^{2^k-1} c_{kj} \chi_{kj}(x) \right| \rightarrow 0 \text{ при } n \rightarrow \infty.$$

Приближением (5.85) функции $f(x)$ обобщенным полиномом n -го порядка по системе функций Хаара является ступенчатая функция

$$g_n(x) = \frac{1}{h} \int_{x_{i-1}}^{x_i} f(x) dx = \text{const} \text{ при } x \in [x_{i-1}; x_i], \quad (5.87)$$

где $x_i = \frac{i}{2^n}$, $h = \frac{1}{2^n}$, $i = 1, \dots, 2^n$.

В качестве примера рассмотрим две задачи.

1. Найти коэффициенты Хаара функции $f(x)$:

$$\begin{aligned} c_{kj} &= \int_0^1 f(x) \chi_{kj}(x) dx = \\ &= 2^{\frac{k-1}{2}} \left[\int_{j_k, 2j-1}^{j_k, 2j} f(x) dx - \int_{j_k, 2j}^{j_k, 2j+1} f(x) dx \right]. \end{aligned} \quad (5.88)$$

Численная реализация этой задачи на микрокалькуляторе осуществляется с помощью программы 24; вычисление определенных интегралов в выражении (5.88) проводится по формуле Симпсона.

2. Найти приближение непрерывной на отрезке $[0; 1]$ функции $f(x)$ функциями Хаара до n -го порядка по формулам (5.87). Вычисления реализуются программой 25.

5.12. Лабораторная работа 8. Интегральная аппроксимация функции методом наименьших квадратов ортогональными полиномами

Программа 23 аппроксимации функции $f(x)$ на отрезке $[-1; 1]$ ортогональными полиномами Лежандра до второго порядка. Интегралы вычисляются по составной формуле Симпсона с шагом $h = 0,5$:

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	2	02	25	2	02	50	×	12
01	F1/X	23	26	÷	13	51	ИП6	66
02	ПД	47	27	Π5	45	52	+	10
03	0	00	28	С/П	50	53	Π6	46
04	Π5	45	29	ΠП	53	54	ИП1	61
05	Π6	46	30	72	72	55	FX ²	22
06	Π7	47	31	ΠП	53	56	3	03
07	1	01	32	44	44	57	×	12
08	/—/	0L	33	ΠП	53	58	1	01
09	Π1	41	34	€8	68	59	—	11
10	ΠП	53	35	ΠП	53	60	2	02
11	29	29	36	42	42	61	÷	13
12	ΠП	53	37	ΠП	53	62	ИП9	69
13	29	29	38	68	68	63	Х	12
14	2	02	39	ΠП	53	64	ИП7	67
15	,	0—	40	44	44	65	+	10
16	4	04	41	В/О	52	66	Π7	47
17	÷	13	42	4	04	67	В/О	52
18	Π7	47	43	×	12	68	ИП1	61
19	ИП6	66	44	Π9	49	69	ИПД	6Г
20	4	04	45	ИП5	65	70	+	10
21	÷	13	46	+	10	71	Π1	41
22	Π6	46	47	Π5	45	72
23	ИП5	65	48	ИП9	69
24	1	01	49	ИП1	61	...	В/О	52

Распределение регистров памяти:

Номер регистра	1	5	6	7	Д	9
Переменная	x	c_0	c_1	c_2	0,5	$f(x)$

Инструкция. 1. Написать подпрограмму вычисления значений функции $f(x)$; значение x хранится в Р1.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В автоматическом режиме (F АВТ) нажать клавиши В/О С/П. После останова прочесть на индикаторе значение c_0 , для прочтения c_1 и c_2 нажать клавиши ИП6 и ИП7 соответственно.

Контрольный пример. Функцию $y = \operatorname{arctg} e^x$ аппроксимировать на отрезке $[-1; 1]$ полиномом второго порядка, выбрав в качестве базиса многочлены Лежандра.

Поставим переключатель Р—Г в положение Р. Составим подпрограмму вычисления значений функции:

ИП1 Fe^X F arctg B/O

После ввода программы в автоматическом режиме нажимаем клавиши В/О С/П. Через 1 мин индицируется $c_0 = 0,78$; нажав клавиши ИП6, прочтем $c_1 = 0,46$, ИП7 — прочтем $c_2 = 0$.

Ответ: $\arctg e^x \approx 0,78 + 0,46x + 0 \cdot \frac{1}{2} (3x^2 - 1)$.

Программа 24 вычисления коэффициентов Хаара до третьего порядка по формулам (5.88):

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	8	08	29	$\frac{+}{-}$	10	58	ПД	4 Γ
01	П0	40	30	ПС	4 \square	59	ИПС	6 \square
02	F1/X	23	31	FL0	5 Γ	60	2	02
03	2	02	32	12	12	61	X	12
04	$\frac{+}{-}$	13	33	С/П	50	62	ПС	4 \square
05	ПД	4 Γ	34	4	04	63	БП	51
06	1	01	35	ПД	4 Γ	64	38	38
07	1	01	36	1	01	65	ИПД	6 Γ
08	П2	42	37	ПС	4 \square	66	ПО	40
09	0	00	38	ИПС	6 \square	67	XU	14
10	ПС	4 \square	39	П1	41	68	КИП2	4 Γ
11	ПВ	4 \square	40	1	01	69	—	11
12	0	00	41	1	01	70	FL0	5 Γ
13	П1	41	42	П2	42	71	68	68
14	ПП	53	43	0	00	72	В/0	52
15	80	80	44	ПП	53	73	ИП1	61
16	ПП	53	45	65	65	74	+	10
17	73	73	46	$/-$	0 \square	75	П1	41
18	4	04	47	ПП	53	76	ИПВ	6 \square
19	X	12	48	65	65	77	ИПД	6 Γ
20	ПП	53	49	ИПС	6 \square	78	+	10
21	73	73	50	FV	21	79	ПВ	4 \square
22	ИП1	61	51	X	12	80
23	+	10	52	С/П	50	...	$f(x)$...
24	4	04	53	FL1	5 \square	...	B/0)	52
25	8	08	54	43	43			
26	$\frac{+}{-}$	13	55	ИПД	6 Γ			
27	КП2	1 \square	56	2	02			
28	ИПС	6 \square	57	$\frac{+}{-}$	13			

Распределение регистров памяти:

Номер регистров	A, 9, ..., 3	В	Д	С	0, 1, 2
Переменная	$\int_{J_1} f(x) dx, \dots, \int_{J_s} f(x) dx$	x	$\Delta x = \frac{1}{16}$	$c_{00}, 2^{k-1}$	счетчики циклов

($k = 1, 2, 3$ — порядок функции Хаара, 2^{3-k} . 2 — количество точек при вычислении интеграла по формуле Симпсона).

Инструкция. 1. Составить подпрограмму вычисления функции $f(x)$, аргумент x хранится в регистре В.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В автоматическом режиме (F АВТ) осуществить пуск программы, нажав клавиши В/О С/П.

4. После каждого из ОСТАНОВов списать с индикатора значение c_{kj} ($c_{01}, c_{11}, c_{21}, c_{22}, c_{31}, c_{32}, c_{33}, c_{34}$) и нажать клавишу С/П.

Контрольный пример. Вычислить коэффициенты Хаара до третьего порядка для функции $f(x) = \exp(-18x^2 + 18x - 4,5)$.

Подпрограмма: ИПВ 6 × 3 — FX² 2 ÷ /—/ Fe^X В/О.

В результате вычислений получим:

$$\begin{aligned}c_{01} &= 4,1663632 \cdot 10^{-1}; & c_{11} &= -4,8 \cdot 10^{-9} \\c_{21} &= -2,1728739 \cdot 10^{-1}; & c_{22} &= 2,1728739 \cdot 10^{-1} \\c_{31} &= -3,648545 \cdot 10^{-2}; & c_{32} &= -9,497388 \cdot 10^{-2} \\c_{33} &= 9,497388 \cdot 10^{-2}; & c_{34} &= 3,648547 \cdot 10^{-2}\end{aligned}$$

Время вычислений c_{01} — около 4 мин, остальных коэффициентов — 15—20 с.

После вычисления коэффициентов можно получить приближение функции $f(x)$ отрезком ряда Хаара $g_3(x)$ по формулам (5.87). Для этого следует умножить значения интегралов, хранящиеся в регистрах А, 9, ..., 3, на 8. Например, вычислим значение $g_3(x)$ на отрезке $[0; 0,125]$: ИПА 8 ×. Прочтем на индикаторе

$$g_3(x) = 3,6374614 \cdot 10^{-2}.$$

Программа 25. Аппроксимация непрерывной функции $f(x)$ на отрезке $[0; 1]$ функциями Хаара n -го порядка по формулам (5.87):

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИП0	60	12	25	25	24	07	07
01	F1/X	23	13	4	04	25	ИПС	6—
02	2	02	14	×	12	26	+	10
03	÷	13	15	ПП	53	27	ПС	4—
04	ПД	4—	16	25	25	28	ИПА	6—
05	0	00	17	ИПС	6—	29	ИПД	6—
06	ПА	4—	18	+	10	30	+	10
07	0	00	19	ПС	4—	31	ПА	4—
08	ПС	4—	20	6	06	32
09	ПП	53	31	÷	13'
10	32	32	22	С/П	50	...	B/0	52
11	ПП	53	23	FL0	5—			

Распределение регистров памяти:

Номер регистра	0	A	C	D
Переменная	2^n	x	Σ	$h/2$

Инструкция. 1. Составить подпрограмму вычисления $f(x)$, значение аргумента x хранится в РА.

2. Ввести программу и подпрограмму в режиме программирования (F ПРГ).

3. В автоматическом режиме (F АВТ) ввести значение 2^n в РО, нажать клавиши В/О С/П.

После первого останова записать с индикатора значение первой константы (5.87) на отрезке $[x_0; x_1]$; нажать клавишу С/П, после останова записать значение второй константы на отрезке $[x_1; x_2]$ и т. д., повторить вычисления 2^n раз.

Контрольный пример тот же, что и в программе 24.

После ввода программы в автоматическом режиме нажимаем клавиши 8 ПО В/О С/П. Примерно через 20 с получим на индикаторе значение $g_3(x) = 3,6374615 \cdot 10^{-2}$, $x \in [0; \frac{1}{8}]$; нажимаем клавишу С/П, после очередного ОСТАНОВа получим:

$$g_3(x) = 1,8231641 \cdot 10^{-1}, x \in \left[\frac{1}{8}; \frac{2}{8} \right); g_3(x) = 5,3397933 \cdot 10^{-1},$$

$$x \in \left[\frac{2}{8}; \frac{3}{8} \right); g_3(x) = 9,1387491 \cdot 10^{-1}, x \in \left[\frac{3}{8}; \frac{4}{8} \right);$$

$$g_3(x) = 9,1387491 \cdot 10^{-1}, x \in \left[\frac{4}{8}; \frac{5}{8} \right);$$

$$g_3(x) = 5,3397933 \cdot 10^{-1}, x \in \left[\frac{5}{8}; \frac{6}{8} \right);$$

$$g_3(x) = 1,8231641 \cdot 10^{-1}, x \in \left[\frac{6}{8}; \frac{7}{8} \right);$$

$$g_3(x) = 3,63374615 \cdot 10^{-2}, x \in \left[\frac{7}{8}; 1 \right].$$

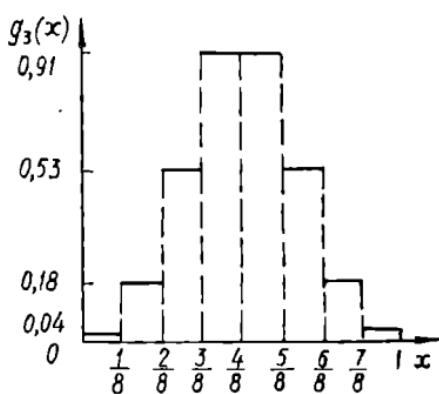


Рис. 13

График полученной аппроксимирующей ступенчатой функции представлен на рис. 13.

Задания к лабораторной работе. А. Функцию $y = f(x)$ аппроксимировать на отрезке $[-1; 1]$ полиномом второго порядка, выбрав в качестве базиса многочлены Лежандра. Переключатель Р—Г поставить в положение Р:

- 1) $y = \frac{1}{\cos x}$;
- 2) $y = \ln(2 + \cos x)$;
- 3) $y = \frac{1}{\ln(x+3)}$;
- 4) $y = \frac{\cos x}{x^2 + 1}$;
- 5) $y = \frac{\sin x}{x^2 + 1}$;
- 6) $y = \frac{\cos x}{2+x}$;
- 7) $y = \frac{x}{\cos x}$;
- 8) $y = \frac{1-x}{\cos x}$;
- 9) $y = \frac{\ln(2+x)}{2+x}$;
- 10) $y = \frac{1}{1+\sin x}$;
- 11) $y = \frac{1}{1+\cos x}$;
- 12) $y = \ln(1+\cos x)$;
- 13) $y = \frac{1}{\ln(3-x)}$;
- 14) $y = \frac{\cos x}{x^2 + 2}$;
- 15) $y = \frac{\sin x}{x^2 + 2}$;
- 16) $y = \frac{\cos x}{x+3}$;
- 17) $y = \frac{2+x}{\cos x}$;
- 18) $y = \frac{1+x}{\cos x}$;
- 19) $y = \sqrt[3]{x^3 + 1}$;
- 20) $y = \frac{1}{\sqrt[3]{x+2}}$;
- 21) $y = \sqrt[4]{x^4 + 1}$;
- 22) $y = \frac{1}{1+\sqrt{x+1}}$;
- 23) $y = \frac{\sqrt{x+1}}{\sqrt{x+2}}$;
- 24) $y = \frac{\sqrt{1-x}}{\sqrt{x+2}}$;
- 25) $y = \frac{\sqrt{1-x}}{\sqrt{2-x}}$;
- 26) $y = \frac{\sqrt{x+1}}{\sqrt{2-x}}$;
- 27) $y = \frac{1}{\sqrt{x^2+1}}$;
- 28) $y = \frac{x}{x^2 + 1}$;
- 29) $y = \sqrt{3+2x}$;
- 30) $y = \frac{1}{\sqrt{3+2x}}$.

Б. Данна функция $f(x) = e^{1-(3x-\frac{n}{10})^2}$, $x \in [0; 1]$, $n \in [1; 30]$, n — номер варианта:

- а) найти ее коэффициенты Хаара до третьего порядка (программа 24);
- б) найти ее приближение отрезком ряда Хаара до третьего порядка (программа 25), построить график найденного приближения.

5.13. Аппроксимация сплайнами

Если функция $f(x)$ задана своими значениями в $(n+1)$ -й точке, то ее можно аппроксимировать интерполяционным полиномом Лагранжа. Но при больших n возникает осцилляция многочлена $P_n(x)$ между узлами, поскольку он имеет точки максимума и минимума в вещественных нулях производной. Устранить это можно уменьшением степени полинома, интерполируя только часть точек. Тогда для всего множества точек получается составная кривая, которая непрерывна на всем интервале, но в узлах «склейки» отдельных кривых ее производные будут разрывны. Погрешность такого

приближения может оказаться значительной. Кроме того, оно неприемлемо в тех практически важных задачах, когда гладкость аппроксимирующей функции играет существенную роль.

Установлено, что интерполяционный многочлен может не стремиться к $f(x)$ при $n \rightarrow \infty$ даже для дифференцируемой произвольное число раз функций [33], например, функции $f(x) = (1 + 25x^2)^{-1}$ на отрезке $[-1; 1]$, для которой

$$\lim_{n \rightarrow \infty} \max_{-1 \leq x \leq 1} |f(x) - P_n(x)| = \infty.$$

Построение полинома невысокой степени для большого числа точек методом наименьших квадратов часто не обеспечивает удовлетворительного соответствия кривой заданным точкам, а повышение степени полинома приводит к существенному усложнению вычислений.

В последнее время в теории аппроксимации широкое применение получили сплайны, позволяющие в некоторой степени устранить недостатки классических методов приближений.

В общем случае сплайн представляет собой функцию, «склеенную» из обобщенных полиномов таким образом, что в точках «склейки» их значения и значения производных до некоторого порядка совпадают. Наиболее простыми являются кусочно-полиномиальные сплайны, для которых в качестве базисных функций берутся алгебраические полиномы.

Гладкость и гибкость сплайнов обуславливает их разнообразное применение в автоматизации проектирования на ЭВМ, в конструировании кривых и поверхностей, в обработке и представлении геометрической информации, а также в самой математике — в теории аппроксимации, при численном интегрировании и дифференцировании, численном решении дифференциальных и интегральных уравнений. Аппарат сплайнов обладает рядом исключительных преимуществ, прежде всего простой реализацией на ЭВМ. При вычислении параметров интерполяционного сплайна, в отличие от интерполяционных полиномов, требуется решение системы алгебраических уравнений с трехдиагональной матрицей. Сплайны обеспечивают высокую точность аппроксимации одновременно и функции, и ее производных. В ряде важных случаев они позволяют получить приближение с минимально возможной погрешностью на данном классе функций по сравнению с другими методами. Следует отметить также устойчивость относительно локальных возмущений сплайна, представленного в виде линейной комбинации базисных B -сплайнов: незначительное изменение значений функции в одном или нескольких узлах

интерполяции мало сказывается на его значениях на некотором удалении от этих точек. Сплайны можно применять для приближения таких функций, степень гладкости которых неодинакова на разных участках аппроксимирующего отрезка. И, наконец, к достоинствам сплайнов следует отнести их простое обобщение на многомерный случай.

5.14. Определение и способы аналитического представления сплайнов

Пусть на отрезке $[a; b]$ задана сетка узлов Δ :

$$a = x_0 < x_1 < x_2 < \dots < x_N = b. \quad (5.89)$$

Обозначим через $C^k [a; b]$ множество функций, непрерывных на $[a; b]$ и имеющих непрерывные производные до k -го порядка, а через P_n — множество многочленов степени не выше n .

Определение 1. Функция $S_n^k(x)$ называется полиномиальным сплайном степени n дефекта k ($1 \leq k \leq n$) с узлами на сетке Δ , если

- 1) $S_n^k(x) \in P_n \forall x \in [x_i; x_{i+1}]$ ($i = 0, 1, \dots, N - 1$);
- 2) $S_n^k(x) \in C^{n-k} [a; b]$.

Дефект k определяет степень гладкости сплайна в узлах и означает, что функция $S_n^k(x)$ и ее производные до $(n - k)$ -го порядка непрерывны на $[a; b]$, а производная порядка $(n - k + 1)$ может быть разрывна в точках x_i ($i = 1, 2, \dots, N - 1$). Если $k = 1$, то для сплайнов дефекта 1 обычно используют обозначение $S_n(x)$.

Будем рассматривать сплайны как аппарат приближения функции $f(x)$, заданной своими значениями на дискретном множестве точек. Для сплайнов нечетной степени узлы интерполяции выбирают, как правило, в узлах сплайна. Пусть на отрезке $[a; b]$ в узлах сетки Δ заданы значения функции $f(x_i) = y_i$ ($i = 0, 1, \dots, N$).

Определение 2. Сплайн $S_{2n+1}^k(x)$ называется интерполяционным, если он удовлетворяет условиям интерполяции:

$$S_{2n+1}^k(x_i) = f(x_i) \quad (i = 0, 1, 2, \dots, N). \quad (5.90)$$

Для однозначного определения параметров сплайна кроме условий интерполяции (5.90) обычно налагаются дополнительные краевые условия. Их будем рассматривать более подробно для случая кубических сплайнов.

Для сплайнов четной степени узлы интерполяции выбирают отличными от узлов сплайна, так как иначе может оказаться, что при некоторых условиях сплайн не существует или же процесс вычисления параметров сплайна будет неустойчивым [28].

Рассмотрим различные способы аналитического представления сплайнов в зависимости от выбора базисных функций.

1. *Кусочно-полиномиальное представление.* Сплайн на каждом отрезке $[x_i; x_{i+1}]$ ($i = 0, 1, \dots, N - 1$) можно представить в виде

$$S_n^k(x) = \sum_{i=0}^n a_{ij} (x - x_i)^j, \quad x \in [x_i; x_{i+1}], \quad (5.91)$$

или

$$S_n^k(x) = \sum_{i=0}^n b_{ij} (x - x_{i+1})^j, \quad x \in [x_i; x_{i+1}]. \quad (5.92)$$

2. *Представление сплайна через усеченную степенную функцию.* Для переменного x и $n \in \mathbb{N}$ введем так называемую усеченную степенную функцию

$$x_+^n = \begin{cases} x^n & \text{при } x \geq 0, \\ 0 & \text{при } x < 0. \end{cases} \quad (5.93)$$

Она является сплайном степени n дефекта 1 с узлом в точке $x = 0$. Рассмотрим усеченную степенную функцию $(x - x_i)_+^n$ (рис. 14), связанную с узлами сетки Δ . Такая функция непрерывна в точке $x = x_i$ вместе со своими производными до порядка $n - 1$, т. е. она является сплайном степени n дефекта 1 с единственным узлом x_i .

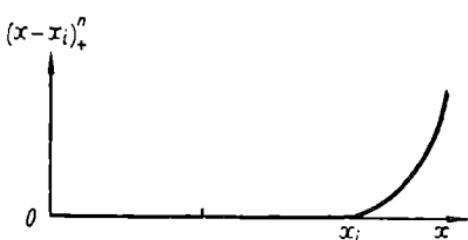


Рис. 14

Функции x^j при $j = 0, 1, 2, \dots, n$ и $(x - x_i)_+^m$ при $m = n - k + 1, \dots, n, i = 1, 2, \dots, N - 1$ линейно независимы на отрезке $[a; b]$ [18]. Поэтому их можно принять в качестве базиса. Получим представление сплайна на отрезке $[a; b]$ в виде усеченных степенных функций

$$\begin{aligned} S(x) \equiv S_n^k(x) = & \sum_{r=0}^n c_r (x - x_0)^r + \\ & + \sum_{i=1}^{N-1} \sum_{j=n-k+1}^n a_{ij} (x - x_i)_+^j, \quad (5.94) \\ & x \in [a; b], \end{aligned}$$

где $c_r = \frac{S^{(r)}(x_0)}{r!}$, $a_{ij} = [S^{(j)}(x_i + 0) - S^{(j)}(x_i - 0)]/j!$ — величины скачков разрывных в узлах производных; $r = 0, 1, 2, \dots, n$; $i = 1, 2, \dots, N - 1$; $j = n - k + 1, \dots, n$.

Такое представление сплайна будет единственным. Из формулы (5.94) и определения усеченной функции (5.93) следует:

$$S(x) = \sum_{r=0}^n c_r (x - x_0)^r \text{ при } x \in [x_0; x_1],$$

$$S(x) = \sum_{r=0}^n c_r (x - x_0)^r + \sum_{i=n-k+1}^n a_{1i} (x - x_1)_+^i \text{ при } x \in [x_0; x_2].$$

Здесь во втором слагаемом (5.94) $i = 1$. При добавлении одного узла сетки Δ количество слагаемых по индексу i в формуле (5.94) увеличивается на единицу.

3. Представление через B -сплайны. Дополним сетку Δ узлами $x_{-n} < x_{-n+1} < \dots < x_{-1} < a$ и $b < x_{N+1} < x_{N+2} < \dots < x_{N+n}$. Часто полагают $x_{i+1} - x_i = x_1 - x_0$ для $i = -2, \dots, -n$ и $x_{j+1} - x_j = x_N - x_{N-1}$ для $j = N, N+1, \dots, N+n-1$. На такой расширенной сетке определим систему $n+N$ функций $\varphi_{ni}(x)$, называемых базисными, или B -сплайнами, полагая их равными разделенным разностям $(n+1)$ -го порядка от функции $\varphi_n(x, t) = (-1)^{n+1}(n+1) \times \times (x-t)_+^n$ по значениям аргумента t в узлах $t = x_i, x_{i+1}, \dots, x_{i+n+1}$ [18].

Получим для B -сплайнов такое выражение:

$$\varphi_{ni} = (n+1) \sum_{j=i}^{i+n+1} \frac{(x_j - x)_+^n}{\omega_{n+1, i}(x_j)}, \quad i = -n, \dots, N-1, \quad (5.95)$$

$$\text{где } \omega_{n+1, i}(x) = \prod_{r=i}^{i+n+1} (x - x_r).$$

Эти функции являются сплайнами степени n дефекта 1 относительно сетки узлов $x_i, x_{i+1}, \dots, x_{i+n+1}$. Приведем следующие свойства B -сплайнов:

- 1) $\varphi_{ni}(x) = 0$ при $x \notin (x_i; x_i + n + 1)$;
- 2) $\varphi_{ni}(x) > 0$ при $x \in (x_i; x_i + n + 1)$;
- 3) $\int_a^b \varphi_{ni}(x) dx = \int_{x_i}^{x_{i+n+1}} \varphi_{ni}(x) dx = 1$.

Функции $\varphi_{ni}(x)$ ($i = -n, -n+1, \dots, N-1$) линейно независимы и образуют базис в пространстве сплайнов сте-

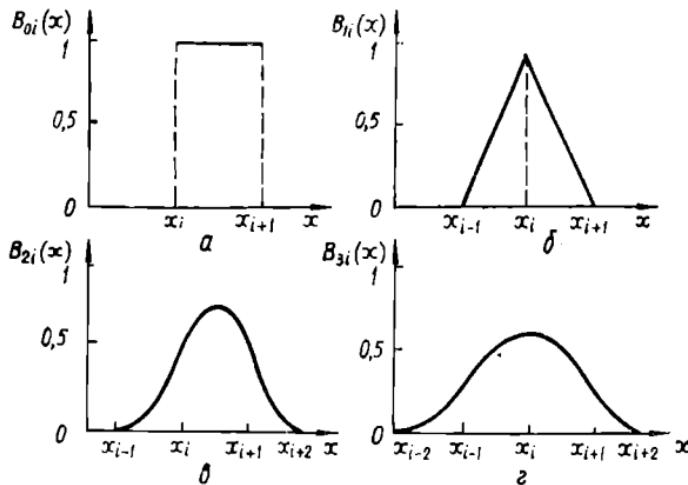


Рис. 15

пени n дефекта 1, размерность которого $n + N$ [18]. Стрезок $[x_i; x_{i+n+1}]$, на котором сплайн $\varphi_{ni}(x)$ отличен от нуля, называется конечным носителем сплайна. Поскольку не существует сплайна степени n дефекта 1 не равного тождественно нулю на интервале, содержащем меньше $n + 1$ частичных отрезков, то $[x_i; x_{i+n+1}]$ называют конечным носителем сплайна минимальной длины.

Из третьего свойства следует, что если узлы x_i и x_{i+n+1} близки, то сплайны $\varphi_{ni}(x)$ принимают значения порядка $O((x_{i+n+1} - x_i)^{-1})$. При их вычислении происходит потеря значащих цифр и сильно увеличивается вычислительная погрешность. Поэтому чаще используют *нормализованные B-сплайны*

$$B_{ni}(x) = \frac{x_{i+n+1} - x}{n+1} \varphi_{ni}(x). \quad (5.96)$$

Вычисление нормализованных *B-сплайнов* можно проводить по рекуррентной формуле [18]

$$B_{ni}(x) = \frac{x - x_i}{x_{i+n} - x_i} B_{n-1, i}(x) + \frac{x_{i+n+1} - x}{x_{i+n+1} - x_{i+1}} B_{n-1, i+1}(x). \quad (5.97)$$

Для равноотстоящих узлов графики *B-сплайнов* нулевой, первой, второй и третьей степеней приведены на рис. 15, $a-g$ соответственно.

Функции $B_{ni}(x)$ ($i = -n, -n + 1, \dots, N - 1$) линейно независимы и образуют базис в пространстве сплайнов степени n дефекта 1. Поэтому любой сплайн $S_n(x)$ на сетке Δ

можно единственным образом представить в виде линейной комбинации $n + N$ нормализованных B -сплайнов

$$S_n(x) = \sum_{i=-n}^{N-1} c_i B_{ni}(x), \quad x \in [a; b], \quad (5.98)$$

где c_i — некоторые постоянные коэффициенты; каждый из сплайнов $B_{ni}(x)$ отличен от нуля на $(n + 1)$ -м последовательном отрезке.

4. *Представление через фундаментальные сплайны.* В теории сплайнов разработаны такие представления интерполяционных сплайнов через линейные комбинации базисных функций, в которых коэффициенты разложения равны значениям интерполируемой функции в узлах или интерполяционным краевым условиям. Пусть заданы значения функции $f(x)$ в узлах x_i сетки Δ : $f(x_i) = y_i$, $i = 0, 1, \dots, N$, а также определенные краевые условия в граничных точках a и b . В общем случае узлы интерполяции могут не совпадать с узлами сплайна. Поэтому кроме сетки узлов Δ рассмотрим другую сетку δ : $-\infty < \xi_1 < \dots < \xi_i < \dots < \xi_{N-n} < +\infty$, узлы которой удовлетворяют условиям

$$x_{i-1} < \xi_i < x_{i+n}, \quad i = 1, 2, \dots, N-n.$$

Рассмотрим систему сплайнов $F_{ni}(x)$ ($i = 0, 1, \dots, N$) степени n дефекта 1, определенных на сетке δ . Каждому узлу x_i сетки Δ поставим в соответствие сплайн $F_{ni}(x)$, удовлетворяющий в узлах интерполяции x_i условиям:

$$F_{ni}(x_j) = \begin{cases} 1 & \text{при } i = j; \\ 0 & \text{при } i \neq j; \end{cases} \quad (5.99)$$

где $x_j \in \Delta$ ($i, j = 0, 1, 2, \dots, N$).

Кроме того, каждому интерполяционному краевому условию сопоставляется сплайн, интерполирующий в этом условии единицу, а во всех остальных — нуль. Сплайны $F_{ni}(x)$ называются *фундаментальными*. При указанном выше выборе узлов сетки δ сплайны $F_{ni}(x)$ существуют и линейно независимы [23].

Любой сплайн степени n дефекта 1, определенный на сетке δ и интерполирующий функцию $f(x)$ в узлах сетки Δ , можно представить в виде

$$S_n(x) = \sum_{i=0}^N y_i F_{ni}(x), \quad x \in [a; b]. \quad (5.100)$$

При наличии в задаче интерполяции краевых условий в формуле (5.100) нужно добавить интерполирующие их фун-

даментальные сплайны. Например, если сплайн $S_n(x)$ удовлетворяет краевым условиям $S'_n(x_0) = y'_0$ и $S'_n(x_N) = y'_N$, то определяем фундаментальные сплайны $F_{ni}(x)$, удовлетворяющие условиям:

$$F_{ni}(x_i) = \delta_{ij} \quad (\delta_{ij} — \text{символ Кронекера}),$$

$$F'_{ni}(x_0) = 0, \quad F'_{ni}(x_N) = 0,$$

дополнительные сплайны $F_{na}(x)$ и $F_{nb}(x)$, удовлетворяющие условиям:

$$F_{na}(x_i) = 0, \quad F'_{na}(x_0) = 1, \quad F'_{na}(x_N) = 0, \quad F_{nb}(x_i) = 0,$$

$$F'_{nb}(x_N) = 1, \quad F_{nb}(x_0) = 0 \quad (i, j = 0, 1, \dots, N).$$

Тогда сплайн $S_n(x)$ можно записать в виде

$$S_n(x) = \sum_{i=0}^N y_i F_{ni}(x) + y'_0 F_{na}(x) + y'_N F_{nb}(x).$$

Формулу (5.100) называют интерполяционной формулой Лагранжа для сплайнов. Если выбрать степень сплайна $n = N$ ($k = 0$), то функции $F_{ni}(x)$ ($i = 0, 1, 2, \dots, N$) будут фундаментальными коэффициентами Лагранжа, а $S_n(x)$ — интерполяционным полиномом Лагранжа. При $n = 1$ и $x_i = \xi_i$ ($i = 1, 2, \dots, N - 1$) получим кусочно-линейную интерполяцию, а сплайн $F_{1i}(x)$ совпадает с B -сплайном первого порядка $B_{1, i-1}(x)$ (см. рис. 15, б). При $n \geq 2$ фундаментальные сплайны $F_{ni}(x)$ имеют более сложный вид (рис. 16).

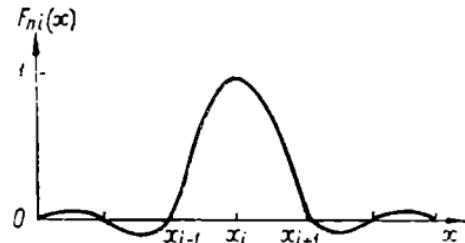


Рис. 16

В практических вычислениях наиболее широкое применение

получили кусочно-полиномиальные представления сплайна (5.91) и (5.92) для каждого частичного отрезка $[x_i; x_{i+1}]$ ($i = 0, 1, \dots, N - 1$), а также аппарат B -сплайнов (5.95) — (5.98). Формула (5.94) используется в основном для представления сплайна на отрезке $[a; b]$ в теоретических исследованиях. Для вычислений она неудобна тем, что при определении параметров c_i и a_{ii} сплайна в задаче интерполяции матрица получаемой при этом системы линейных уравнений может оказаться плохо обусловленной. Кроме того, усеченные степенные функции $(x - x_i)_+^n$ при $x > x_i$ быстро возрастают при удалении x от x_i , что приводит к накоплению вычислительной погрешности. В представлении сплайна через фунда-

ментальные сплайны (5.100) предельно просто вычисляются параметры сплайна, но сама процедура вычисления фундаментальных сплайнов $F_{ni}(x)$ является достаточно громоздкой [23].

5.15. Интерполяция кубическими сплайнами

В инженерной практике наиболее широкое применение получили интерполяционные кубические сплайны. Это обусловлено тем, что полином третьей степени является простейшей кривой, имеющей точку перегиба, что обеспечивает его хорошие интерполяционные возможности. Кубические сплайны имеют на всем отрезке аппроксимации непрерывные производные до второго порядка. Такая гладкость обычно оказывается достаточной для большинства задач. Невысокая степень полинома упрощает вычисления и уменьшает вычислительную погрешность.

Пусть функция $f(x)$ задана на отрезке $[a; b]$ в узлах сетки Δ : $a = x_0 < x_1 < x_2 < \dots < x_N = b$ значениями $f(x_i) = y_i$ ($i = 0, 1, \dots, N$). Будем рассматривать сплайны третьей степени дефекта 1.

Определение. Функция $S_3(x)$ называется кубическим сплайном, интерполирующим функцию $f(x)$ в узлах сетки Δ , если:

1) на каждом из отрезков $[x_i; x_{i+1}]$ ($i = 0, 1, \dots, N - 1$) она является многочленом не выше третьей степени, т. е. $S_3(x) \in P_3$;

2) $S_3(x)$ непрерывна на $[a; b]$ вместе со своими производными до второго порядка включительно, т. е. $S_3(x) \in C^2[a; b]$;

3) в узлах сетки функция $S_3(x)$ удовлетворяет условиям интерполяции

$$S_3(x_i) = f(x_i) \quad (i = 0, 1, \dots, N). \quad (5.101)$$

Воспользуемся кусочно-полиномиальным представлением (5.91) кубического сплайна на отрезке $(x_i; x_{i+1})$:

$$S_3(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3, \quad (5.102)$$

$$x \in [x_i; x_{i+1}].$$

На каждом частичном отрезке $[x_i; x_{i+1}]$ кубический сплайн определяется четырьмя параметрами, а на всем отрезке $[a; b]$ число параметров равно $4N$. Для их определения нужно воспользоваться $(N + 1)$ -м условием интерполяции (5.101), а также условиями непрерывности сплайна и его первой и второй производных во внутренних узлах x_1, x_2, \dots, x_{N-1} сетки, число которых равно $3(N - 1)$. Получим $4N - 2$ условия для опре-

деления $4N$ неизвестных параметров. Недостающие для однозначного определения сплайна два условия задаются дополнительно в виде ограничений на значения сплайна или его производных на концах отрезка $[a; b]$ в зависимости от физических условий каждой конкретной задачи. Наиболее часто применяют следующие краевые условия:

$$1) \quad S'_3(a) = f'(a), \quad S'_3(b) = f'(b);$$

$$2) \quad S''_3(a) = f''(a), \quad S''_3(b) = f''(b);$$

$$3) \quad S^{(r)}_3(a) = S^{(r)}_3(b), \quad r = 1, 2 \text{ — условие периодичности};$$

4) $S'''_3(x_i - 0) = S'''_3(x_i + 0)$, $i = 1, N - 1$ — дополнительные условия непрерывности, где $S'''_3(x_i + 0)$ и $S'''_3(x_i - 0)$ — соответственно правая и левая односторонние производные третьего порядка в точке x_i .

Краевые условия 3 применяют для интерполяции периодических функций периода $T = b - a$. Кубический сплайн, удовлетворяющий краевым условиям 2, где $f''(a) = 0$ и $f''(b) = 0$ (нулевая кривизна), называется естественным.

Рассмотрим алгоритмы вычисления параметров кубического сплайна (5.102) при краевых условиях 1—4. Введем обозначения:

$$\begin{aligned} h_i &= x_{i+1} - x_i \quad (i = 0, 1, \dots, N - 1), \quad M_i = S''_3(x_i), \\ m_i &= S'_3(x_i) \quad (i = 0, 1, \dots, N). \end{aligned}$$

Величины M_i и m_i называют моментами и наклонами сплайна в узле x_i соответственно.

Так как $S_3(x)$ — кубическая функция, то $S''_3(x)$ будет линейной относительно x функцией на каждом частичном отрезке. Поэтому при $x \in [x_i; x_{i+1}]$ по формуле линейной интерполяции (5.14) запишем выражение для второй производной сплайна в виде

$$S''_3(x) = S''_3(x_i) + \frac{S''_3(x_{i+1}) - S''_3(x_i)}{x_{i+1} - x_i} (x - x_i),$$

или через моменты

$$S''_3(x) = M_i + \frac{M_{i+1} - M_i}{h_i} (x - x_i). \quad (5.103)$$

Проинтегрируем дважды это равенство:

$$S'_3(x) = S'_3(x_i) + M_i(x - x_i) + \frac{M_{i+1} - M_i}{2h_i} (x - x_i)^2; \quad (5.104)$$

$$S_3(x) = y_i + S'_3(x_i)(x - x_i) + \frac{M_i}{2} (x - x_i)^2 + \frac{M_{i+1} - M_i}{6h_i} (x - x_i)^3. \quad (5.105)$$

Полагаем в последнем равенстве $x = x_{i+1}$:

$$y_{i+1} = y_i + S'_3(x_i) h_i + \frac{M_i}{2} h_i^2 + \frac{M_{i+1} - M_i}{6h_i} h_i^3,$$

откуда

$$S'_3(x_i) = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}).$$

Подставим значение $S'_3(x_i)$ в формулы (5.104) и (5.105):

$$\begin{aligned} S_3(x) &= y_i + \left[\frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}) \right] (x - x_i) + \\ &+ \frac{M_i}{2}(x - x_i)^2 + \frac{M_{i+1} - M_i}{6h_i}(x - x_i)^3, \quad x \in [x_i, x_{i+1}] \\ (i &= 0, 1, \dots, N-1); \end{aligned} \quad (5.106)$$

$$\begin{aligned} S'_3(x) &= \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}) + M_i(x - x_i) + \\ &+ \frac{M_{i+1} - M_i}{2h_i}(x - x_i)^2, \quad x \in [x_i, x_{i+1}] \quad (i = 0, 1, \dots, N-1). \end{aligned} \quad (5.107)$$

Формулы (5.106) и (5.107) выражают интерполяционный кубический сплайн и его производную на отрезке $[x_i; x_{i+1}]$ через значения функции $f(x)$ в узлах и моменты M_i, M_{i+1} .

Учитывая равенства (5.102) и (5.106), запишем выражения для параметров сплайна в кусочно-полиномиальном представлении

$$\begin{cases} a_{i1} = y_i, \quad a_{i1} = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}), \\ a_{i2} = \frac{M_i}{2}, \quad a_{i3} = \frac{M_{i+1} - M_i}{6h_i} \quad (i = 0, 1, \dots, N-1). \end{cases} \quad (5.108)$$

Для определения моментов M_i, M_{i+1} воспользуемся свойством непрерывности первой производной $S'_3(x)$ в узлах сплайна. Заменим в (5.107) индекс i на $i-1$, получим выражение $S'_3(x)$ на отрезке $[x_{i-1}; x_i]$:

$$\begin{aligned} S'_3(x) &= \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{6}(2M_{i-1} + M_i) + \\ &+ M_{i-1}(x - x_{i-1}) + \frac{M_i - M_{i-1}}{2h_{i-1}}(x - x_{i-1})^2. \end{aligned} \quad (5.109)$$

Из равенств $S'_3(x_i - 0) = S'_3(x_i + 0)$ следует:

$$\begin{aligned} h_{i-1}M_{i-1} + 2(h_i + h_{i-1})M_i + h_iM_{i+1} &= \frac{6(y_{i+1} - y_i)}{h_i} - \frac{6(y_i - y_{i-1})}{h_{i-1}}, \\ i &= 1, 2, \dots, N-1. \end{aligned}$$

Последнее выражение представим в виде

$$\alpha_i M_{i-1} + 2M_i + \beta M_{i+1} = d_i, \quad (5.110)$$

где $\alpha_i = \frac{h_{i-1}}{h_i + h_{i-1}}$, $\beta_i = \frac{h_i}{h_i + h_{i-1}} = 1 - \alpha_i$,

$$d_i = \frac{6}{h_i + h_{i-1}} \left[\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right] = 6f(x_{i-1}; x_i; x_{i+1}) \quad (5.111)$$

$$(i = 1, 2, \dots, N-1).$$

Для определения моментов M_i во всех узлах x_i ($i = 0, 1, \dots, N$) прибавим к (5.110) еще два уравнения, вытекающие из соответствующих краевых условий:

$$1) 2M_0 + M_1 = d_0, \text{ где } d_0 = \frac{6}{h_0} \left(\frac{y_1 - y_0}{h_0} - f'(a) \right), \quad (5.112)$$

$$M_{N-1} + 2M_N = d_N, \text{ где } d_N = \frac{6}{h_{N-1}} \left(f'(b) - \frac{y_N - y_{N-1}}{h_{N-1}} \right);$$

$$2) M_0 = f''(a), M_N = f''(b); \quad (5.113)$$

$$3) 2M_1 + \beta_1 M_2 + \alpha_1 M_N = d_1,$$

$$\beta_N M_1 + \alpha_N M_{N-1} + 2M_N = d_N,$$

$$M_0 = M_N; \quad (5.114)$$

$$4) (1 + \beta_1) M_1 + (\beta_1 - \alpha_1) M_2 = \beta_1 \alpha_1,$$

$$(\alpha_{N-1} - \beta_{N-1}) M_{N-2} + (1 + \alpha_{N-1}) M_{N-1} = \alpha_{N-1} d_{N-1} \quad (5.115)$$

$$M_0 = \beta_1^{-1} (M_1 - \alpha_1 M_2), M_N = \alpha_{N-1}^{-1} (M_{N-1} - \beta_{N-1} M_{N-2}),$$

в (5.110) полагаем $i = 2, 3, \dots, N-2$.

В случае краевых условий 3 и 4 преобразованы первое и последнее уравнения системы и из них исключены M_0 и M_N . В результате получены системы линейных алгебраических уравнений, у каждой из которых матрица имеет доминирующую главную диагональ. Такая матрица будет невырожденной [19], поэтому система (5.110), дополненная соответственно уравнениями (5.112)–(5.115), имеет единственное решение. Это условие свидетельствует также о существовании и единственности интерполяционного сплайна.

Таким образом, задача построения интерполяционного кубического сплайна сводится к решению системы алгебраических уравнений с трехдиагональной матрицей, что можно эффективно осуществить методом прогонки. Для матрицы с диагональным преобладанием метод прогонки устойчив относительно ошибок округления.

Если моменты M_i ($i = 0, 1, \dots, N$) во всех узлах вычислены, то кубический сплайн на каждом частичном отрезке $[x_i; x_{i+1}]$ ($i = 0, 1, \dots, N-1$) представляем в виде (5.106)

или (5.102). Выражение (5.106) для сплайна можно преобразовать к более симметричному виду

$$S(t) = (1-t)y_i + ty_{i+1} - \frac{h_i^2}{6} t(1-t)[(2-t)M_i + (5.116) \\ + (1+t)M_{i+1}], \quad t = \frac{x-x_i}{h_i}, \quad x \in [x_i; x_{i+1}] \quad (i=0, 1, \dots, N-1).$$

При кусочно-многочленном представлении сплайна требуется запоминать узлы сетки x_i , значения функции y_i и моменты M_i ($i=0, 1, \dots, N$).

Для вычисления значения кубического сплайна в точке $x \in [a; b]$ можно воспользоваться формулой (5.102) и схемой Горнера

$$S_3(x) = ((a_{i3}(x-x_i) + a_{i2})(x-x_i) + a_{i1})(x-x_i) + a_{i0}. \quad (5.117)$$

Коэффициенты a_{ij} ($j=0, 1, 2, 3$) согласно (5.108) удобно вычислять по схеме

$$a_{i0} = y_i, \quad a_{i2} = \frac{M_i}{2}, \quad a_{i3} = \frac{M_{i+1} - M_i}{6h_i}, \quad (5.118) \\ a_{i1} = \frac{y_{i+1} - y_i}{h_i} - h_i(a_{i2} + a_{i3}h_i).$$

Предварительно следует определить интервал $[x_i; x_{i+1}]$, которому принадлежит указанное значение аргумента.

В некоторых задачах более удобно представление кубического сплайна через наклоны $m_i = S'_3(x_i)$, $i=0, 1, 2, \dots, N$. Воспользуемся формулой (5.21) для интерполяционного многочлена Эрмита с двумя узлами x_i и x_{i+1} кратности 2, выражающей интерполяционный кубический многочлен на отрезке $[x_i; x_{i+1}]$ через значения функции y_i и ее производных m_i в граничных точках отрезка. Получим выражение кубического сплайна на каждом частичном отрезке через наклоны

$$S_3(x) = y_i + (x-x_i)m_i + \frac{(x-x_i)^2}{h_i} \left(\frac{y_{i+1}-y_i}{h_i} - m_i \right) - \\ - \left(\frac{x-x_i}{h_i} \right)^2 (x_{i+1}-x) \left(m_{i+1} + m_i - 2 \frac{y_{i+1}-y_i}{h_i} \right); \quad (5.119) \\ x \in [x_i; x_{i+1}] \quad (i=0, 1, \dots, N-1).$$

Сгруппируем коэффициенты при неизвестных значениях наклонов. Получим представление сплайна в другой форме

$$S_3(x) = \frac{(x_{i+1}-x)^2(x-x_i)}{h_i^2} m_i - \frac{(x-x_i)^2(x_{i+1}-x)}{h_i^2} m_{i+1} + (5.120) \\ + \frac{(x_{i+1}-x)^2(2(x-x_i)+h_i)}{h_i^3} y_i + \frac{(x-x_i)^2(2(x_{i+1}-x)+h_i)}{h_i^3} y_{i+1}.$$

Из формулы (5.119) найдем выражение для второй производной кубического сплайна

$$\begin{aligned} S_3''(x) &= \frac{2}{h_i} \left(\frac{y_{i+1} - y_i}{h_i} - m_i \right) - \left(\frac{2(x_{i+1} - x)}{h_i^2} - \right. \\ &\quad \left. - \frac{4(x - x_i)}{h_i^3} \right) \left(m_i + m_{i-1} - 2 \frac{y_{i+1} - y_i}{h_i} \right) = \quad (5.121) \\ &= (y_{i+1} - y_i) \frac{6 - 12t}{h_i^2} + m_i \frac{(-4 + 6t)}{h_i} + m_{i+1} \frac{(-2 + 6t)}{h_i}, \end{aligned}$$

где $t = \frac{x - x_i}{h_i}$, $x \in [x_i; x_{i+1}]$ ($i = 0, 1, 2, \dots, N - 1$).

Полагая в (5.121) $x = x_i$, вычислим правую одностороннюю производную второго порядка в узле x_i :

$$S_3''(x_i + 0) = -\frac{4m_i}{h_i} - \frac{2m_{i+1}}{h_i} + 6 \frac{y_{i+1} - y_i}{h_i^2}.$$

Заменим в (5.121) индекс i на $i - 1$ и вычислим левую одностороннюю производную второго порядка в узле x_i :

$$S_3''(x_i - 0) = \frac{2m_{i-1}}{h_{i-1}} + \frac{4m_i}{h_{i-1}} - 6 \frac{y_i - y_{i-1}}{h_{i-1}^2}.$$

По условию непрерывности второй производной кубического сплайна в узлах x_i полагаем

$$S_3''(x_i - 0) = S_3''(x_i + 0) \quad (i = 1, 2, \dots, N - 1),$$

что эквивалентно системе линейных алгебраических уравнений относительно наклонов

$$\beta_i m_{i-1} + 2m_i + \alpha_i m_{i+1} = c_i \quad (i = 1, 2, \dots, N - 1), \quad (5.122)$$

где $\alpha_i = \frac{h_{i-1}}{h_i + h_{i-1}}$, $\beta_i = \frac{h_i}{h_i + h_{i-1}} = 1 - \alpha_i$, $c_i = 3 \left(\alpha_i \frac{y_{i+1} - y_i}{h_i} + \beta_i \frac{y_i - y_{i-1}}{h_{i-1}} \right)$.

К уравнениям (5.122) добавим недостающие два уравнения в соответствии с заданными краевыми условиями типа:

$$1) \quad m_0 = f'(a), \quad m_N = f'(b); \quad (5.123)$$

$$2) \quad 2m_0 + m_1 = 3 \frac{y_1 - y_0}{h_0} - \frac{h_0}{2} f''(a),$$

$$m_{N-1} + 2m_N = 3 \frac{y_N - y_{N-1}}{h_{N-1}} + \frac{h_{N-1}}{2} f''(b); \quad (5.124)$$

$$3) \quad 2m_1 + \alpha_1 m_2 + \beta_1 m_N = c_1,$$

$$\alpha_N m_1 + \beta_N m_{N-1} + 2m_N = c_N, \quad (5.125)$$

где по условию периодичности $y_0 = y_N$, $y_1 = y_{N+1}$, $m_0 = m_N$, $m_1 = m_{N+1}$, $h_N = h_0$, в системе уравнений (5.122) полагаем $i = 2, 3, \dots, N-1$;

$$4) \left(1 + \frac{h_0}{h_1}\right)m_1 + \frac{h_0}{h_1}m_2 = \frac{1}{3}c_1 + 2\frac{h_0}{h_1} \cdot \frac{y_2 - y_1}{h_1},$$

$$\frac{h_{N-1}}{h_{N-2}}m_{N-2} + \left(1 + \frac{h_{N-1}}{h_{N-2}}\right)m_{N-1} = \frac{1}{3}c_{N-1} +$$

$$+ 2\frac{h_{N-1}}{h_{N-2}}\frac{y_{N-1} - y_{N-2}}{h_{N-2}}. \quad (5.126)$$

В системе уравнений (5.122) полагаем $i = 2, 3, \dots, N-2$.

Для краевых условий всех типов система алгебраических уравнений (5.122), дополненная соответственно уравнениями (5.123)–(5.126), имеет матрицу с доминирующей диагональю, поэтому существует единственное решение, которое может быть получено методом прогонки.

Для вычисления значений сплайна $S_3(x)$ в точке $x \in [a; b]$ при известных значениях наклонов m_i ($i = 0, 1, \dots, N$) можно воспользоваться формулой (5.119), преобразованной к виду, удобному для вычислений на микрокалькуляторе:

$$S_3(t) = y_i + (x - x_i)(m_i + t(R + tA)), \quad (5.127)$$

где

$$A = -2\frac{y_{i+1} - y_i}{h_i} + m_{i+1} + m_i;$$

$$B = -A + \frac{y_{i+1} - y_i}{h_i} - m_i; \quad t = \frac{x - x_i}{h_i}.$$

Как уже отмечалось, сплайны обладают хорошими аппроксимативными свойствами. Интерполяционные сплайны в ряде задач обеспечивают минимально возможную погрешность приближения на данном классе функций среди всех многочленов фиксированной степени. Так, установлено [1], что среди всех функций, интерполирующих $f(x)$ в заданных точках и имеющих интегрируемую с квадратом вторую производную на $[a; b]$, кубический сплайн является единственной функцией, обладающей свойством минимальной кривизны, т. е. обеспечивающей минимум интеграла $\int_a^b [f''(x)]^2 dx$. В этом смысле кубический сплайн есть самая гладкая из функций, интерполирующих $f(x)$.

В настоящее время получены многочисленные результаты по точным или асимптотически точным оценкам погрешности

сплайн-интерполяции на классах функций [18, 23, 28]. Ограничимся здесь только порядковыми оценками погрешности интерполяции кубическими сплайнами. Если интерполируемая функция $f(x)$ принадлежит классу $C^k[a; b]$ функций, непрерывных на $[a; b]$ и имеющих непрерывные производные до k -го порядка ($k = 0, 1, 2, 3, 4$), то для погрешности интерполяции кубическим сплайном $S_3(x)$ функции и ее производных действительны оценки.

$$\max_{a \leq x \leq b} |f^{(p)}(x) - S_3^{(p)}(x)| \leq ch^{k-p}, \quad k \geq p, \quad (5.128)$$

где c — неотрицательная константа, не зависящая от сетки Δ ;

$$h = \max_{0 \leq i \leq N-1} |x_{i+1} - x_i|.$$

Для функции $f(x) \in C^4[a; b]$ порядок приближения $f(x)$ сплайном $S_3(x)$ равен $O(h^4)$, а порядок приближения производных $f'(x)$ и $f''(x)$ функциями $S_3'(x)$ и $S_3''(x)$ равен соответственно $O(h^3)$ и $O(h^2)$, т. е. одновременно получаются хорошие приближения и для производных.

Точность приближения интерполяционными сплайнами зависит, кроме гладкости функции, от выбора сетки Δ , вида краевых условий, а также степени сплайна. В связи с этим целесообразно точки разрыва производных интерполируемой функции включать в число узлов сплайна. Степень сплайна должна быть согласована с гладкостью интерполируемой функции $f(x)$, так как при повышении степени сплайна погрешность приближения может ухудшаться. Обычно требуется некоторая априорная информация о дифференцируемости приближаемой функции $f(x)$, ее поведении на концах отрезка $[a; b]$. Если, например, заданы только значения функции в узлах сетки Δ и ничего не известно о выборе краевых условий, то часто функцию приближают естественным кубическим сплайном, положив $f''(a) = f''(b) = 0$. При этом точность приближения в точках, близких к концам отрезка $[a; b]$, снижается и становится почти равной точности приближения сплайном первой степени. Такую потерю точности иногда можно ослабить, применив разностную аппроксимацию производных функции $f(x)$ в граничных точках. Чаще всего выбирают краевые условия типа 4.

Отметим также, что кубические сплайны обладают в определенной мере свойством локальности, заключающимся в следующем: изменение значения функции в узле x_i мало влияет на значения сплайна и его производных в точках, достаточно удаленных от узла x_i [18].

При определении параметров кубического сплайна на микрокалькуляторе будем выбирать равномерную сетку с шагом $h = \frac{b-a}{N} = \text{const.}$

Пусть интерполяционный кубический сплайн удовлетворяет на концах отрезка $[a; b]$ краевым условиям типа I. Если сплайн выразить через моменты $M_i = S''(x_i)$ ($i = 0, 1, \dots, N$) по формуле (5.106) или (5.116), то для определения параметров M_i , учитывая (5.110)–(5.112), нужно решить систему уравнений

$$\left| \begin{array}{ccccccc|c|c|c} 1 & \frac{1}{2} & 0 & 0 & \cdots & 0 & 0 & 0 & M_0 & d_0 \\ \frac{1}{2} & 2 & \frac{1}{2} & 0 & \cdots & 0 & 0 & 0 & M_1 & d_1 \\ 0 & \frac{1}{2} & 2 & \frac{1}{2} & \cdots & 0 & 0 & 0 & M_2 & d_2 \\ \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{2} & 2 & \frac{1}{2} & M_{N-1} & d_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{2} & 1 & M_N & d_N \end{array} \right| = \left| \begin{array}{c} d_0 \\ d_1 \\ d_2 \\ \cdots \\ d_{N-1} \\ d_N \end{array} \right|$$

Здесь

$$d_i = \frac{3}{h^2} (y_{i-1} - 2y_i + y_{i+1}) \quad (i = 0, 1, \dots, N);$$

$$y_{-1} = y_0 - hf'(a), \quad y_{N+1} = y_N + hf'(b).$$

Вычисление значений d_i , стоящих в правой части, а также решение системы методом прогонки в случае, если число узлов интерполяции не превышает 9, реализуется программой 26.

Программа 26 вычисления моментов M_i ($i = 0, 1, \dots, N$) кубического сплайна, интерполирующего таблицу значений с постоянным шагом h и удовлетворяющего краевым условиям типа 1, $0 \ll N \ll 8$;

ИПД	4	+	ПО	КИПО	ПВ	КИПО	ПА	КИПО	ИПВ	
ИПА	ПВ	2	\times	КИП	\uparrow	ПА	$-$	$-$	ИПС	FX ²
\div	3	\times	КП	\uparrow	FLO	09	ИПД	ПО	1	ПС
ПА	2	ПВ	КИПС	ИПА	ИПВ	\div	\times	ИПС	1	
$+$	ПС	F,	КИПС	ХY	$-$	КПС	ПП	82	FLO	
33	ИПД	1	$+$	ПО	ИПВ	ИПА	ПВ	ХY	ПА	
ИПВ	\div	2	\div	1	$-$	КИП	\uparrow	ХY	\div	КП
С/П	КИПО	\uparrow	$+$	ХY	$-$	ПП	82	\times	ИПА	
БП	68	ИПА	ИПВ	ПА	4	\times	ХY	$-$	ПВ	
B/O										

Распределение регистров памяти:

Номер регистра	0	1, 2, ..., В	с	Д
Переменная индекс цикла		$y_{-1}, y_0, \dots, y_{N+1}$ d_0, d_1, \dots, d_N m_0, m_1, \dots, m_N	h	N

Инструкция. 1. Вычислить $y_{-1} = y_0 - hf'(a)$ и $y_{N+1} = y_N + hf'(b)$.

2. В режиме программирования (F ПРГ) ввести программу.

3. В режиме автоматической работы (F АВТ) ввести исходные данные:

(y_{-1}) П1 (y_0) П2 ... (y_{N+1}) П ($N + 3$) (h) ПС (N) ПД

4. Приступить к вычислениям, нажав клавиши В/О С/П. После первого ОСТАНОВА прочесть на индикаторе M_N . Нажать клавишу С/П и после каждого очередного ОСТАНОВА прочесть на индикаторе M_i ($i = N - 1, \dots, 1, 0$).

После окончания вычислений значения M_i ($i = 0, 1, \dots, N$) хранятся в регистрах 1, 2, ..., $N + 1$ соответственно.

Контрольный пример. Найти параметры M_i кубического сплайна $S_3(x)$, интерполирующего функцию $f(x) = e^x$ на отрезке $[0; 0,5]$, шаг $h = 0,1$ $N = 5$. Исходные данные:

i	0	1	2	3	4	5
x_i	0	0,1	0,2	0,3	0,4	0,5
y_i	1	1,10517	1,22140	1,34986	1,49182	1,64872
M_i	0,99902	1,10395	1,22117	1,34938	1,48132	1,68934

Зададим краевые условия: $f'(0) = 1$; $f'(0,5) = 1,65$. Вычисляем $y_{-1} = 1 - 0,1 \cdot 1 = 0,9$; $y_6 = 1,64872 + 0,1 \cdot 1,65 = 1,81372$. После ввода программы, нажав клавиши F АВТ, вводим исходные данные:

0,9 П1 1 П2 1,10517 П3 1,22140 П4 1,34986 П5 1,49182
П6 1,64872 П7 1,81372 П8 0,1 ПС 5 ПД

Осуществляем пуск программы, нажав клавиши В/О С/П. После ОСТАНОВА (примерно через 1,5 мин) читаем на индикаторе $M_5 = 1,68934$. Нажимая клавишу С/П, после каждого ОСТАНОВА записываем в таблицу справа налево значения M_i ($i = 4, 3, 2, 1, 0$).

Для вычисления значения кубического сплайна в произвольной точке $x \in [x_i; x_{i+1}]$ по найденным значениям наклонов M_i ($i = 0, 1, \dots, N$) определяем коэффициенты сплайна a_{ij} ($j = 0, 1, 2, 3$):

$$a_{i0} = y_i, \quad a_{i2} = \frac{M_i}{2}, \quad a_{i3} = \frac{M_{i+1} - M_i}{6h}, \quad a_{i1} = \frac{y_{i+1} - y_i}{h} - h(a_{i2} + a_{i3}h).$$

Затем по схеме Горнера вычисляем

$$S_3(x) = ((a_{i3}(x - x_i) + a_{i2})(x - x_i) + a_{i1})(x - x_i) + a_{i0}.$$

Вычисление сплайна в произвольной точке $x \in [a; b]$ реализуется программой 27, которую можно вводить в память микрокалькулятора с адреса 00 после выполнения программы 26, не изменяя содержимого его регистров.

Программа 27 вычисления по формулам (5.118), (5.117) значения интерполяционного кубического сплайна в точке $x \in [x_i; x_{i+1}]$ по известным значениям наклонов M_i ($i = 0, 1, \dots, N$) и табличным значениям интерполируемой функции y_i с постоянным шагом h , $0 \leq N \leq 8$:

КИП↑ ↑ КИПО — 6 ÷ ИПС ÷ ИПД ×
КИП↑ 2 ÷ + ИПД × XY КИП↑ ↑ +
+ 6 ÷ ИПС × — ИПВ ИПА — ИПС
÷ + ИПД × ИПА + С/П

Распределение регистров памяти:

Номер регистра	0	1, 2, ..., 9	A	B	C	Д
Переменная	$i + 2$	M_0, M_1, \dots, M_8	y_i	y_{i+1}	h	$\Delta x_i = x - x_i$

Инструкция. 1. Если проводились вычисления по программе 26, то не отключать микрокалькулятор. Значения моментов M_i ($i = 0, 1, \dots, N$) хранятся в регистрах 1, 2, ..., $N + 1$.

2. В режиме программирования (F ПРГ) ввести программу с адреса 00.

3. В режиме автоматической работы (F АВТ) ввести исходные данные: $(i + 2)$ ПО (y_i) ПА (y_{i+1}) ПВ (h) ПС (x) ↑ (x_i) — ПД; и, если не выполнено условие п. 1, (M_i) П ($i + 1$) (M_{i+1}) П ($i + 2$).

4. Нажать клавиши В/О С/П. После ОСТАНОВа прочесть на индикаторе ответ.

По результатам предыдущего контрольного примера вычислим значение кубического сплайна, интерполирующего функцию $f(x) = e^x$ на отрезке $[0; 0,5]$ в точке $x = 0,25$. Здесь $i = 2$; $x_i = 0,2$; $y_i = 1,22140$, $y_{i+1} = 1,34986$. После выпол-

нения указаний инструкции получим $S_3(0,25) = 1,2840234$. Точное значение функции $f(0,25) = e^{0,25} = 1,2840254\dots$, т. е. в данном примере пять знаков после запятой в аппроксимирующем сплайне — верные цифры. Время вычислений значения сплайна в одной точке — примерно 12 с.

5.16. Кубические B -сплайны

Рассмотрим задачу сплайн-интерполяции функции $f(x)$, заданной своими значениями $y_i = f(x_i)$ в узлах сетки $\Delta a = x_0 < x_1 < \dots < x_N = b$. Для представления сплайна $S_3(x)$ на отрезке $[a; b]$ в качестве базисных функций выберем кубические B -сплайны. Дополним сетку Δ на каждом конце тремя узлами $x_{-3} < x_{-2} < x_{-1} < a; b < x_{N+1} < x_{N+2} < x_{N+3}$. Для периодического сплайна полагаем $h_{N+i} = h_i, i = 0, 1, 2$ ($h_i = x_{i+1} - x_i$); в общем случае можно положить $h_{-2} = h_{-1} = h_0$ и $h_{N+1} = h_N = h_{N-1}$.

Обозначим через $\varphi_i(x)$ кубический B -сплайн, определенный на интервале $[x_{i-2}; x_{i+2}]$ с центром в точке x_i . Индекс кубического сплайна выберем по среднему узлу интервала-носителя. Функция $\varphi_i(x)$ является кубическим многочленом дефекта 1 по переменной x на сетке узлов $x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}$. Такой сплайн можно определить через разделенную разность 4-го порядка для функции $\varphi(x, y) = 4(y - x)_+^3$ по переменной y :

$$\varphi_i(x) = \varphi(x; x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}) = 4 \sum_{k=i-2}^{i+2} \frac{(x_k - x)_+^3}{\omega_i'(x_k)}, \quad (5.129)$$

где

$$\omega_i(x) = \prod_{k=i-2}^{i+2} (x - x_k).$$

При $x \geq x_{i+2} \varphi_i(x) \equiv 0$ по определению усеченной степенной функции. При $x \leq x_{i-2}$ функция $\varphi(x, y)$ будет обычным кубическим полиномом по переменной $y \in (x_{i-2}; x_{i+2})$, а ее разделенная разность 4-го порядка равна нулю. Таким образом, кубический B -сплайн является финитной функцией, т. е. он тождественно равен нулю на всей числовой оси, кроме точек интервала-носителя:

$$\varphi_i(x) \equiv 0 \text{ при } x \notin (x_{i-2}; x_{i+2}). \quad (5.130)$$

Отметим также, что для B -сплайна выполняются условия [18]:

$$\varphi_i(x) > 0 \text{ при } x \in (x_{i-2}; x_{i+2}); \quad (5.131)$$

$$\int_a^b \varphi_i(x) dx = \int_{x_{i-2}}^{x_{i+2}} \varphi_i(x) dx = 1. \quad (5.132)$$

Нормализованные кубические B -сплайны определяются по формуле

$$B_i(x) = \frac{x_{i+2} - x_{i-2}}{4} \varphi_i(x). \quad (5.133)$$

На расширенной сетке построим систему кубических сплайнов $B_i(x)$, $i = -1, 0, 1, \dots, N, N + 1$. Условие нормировки (5.132) для нормализованных сплайнов будет иметь вид

$$\sum_{i=-1}^{N+1} B_i(x) = 1. \quad (5.134)$$

Значение сплайна $B_i(x)$ ($i = -1, 0, 1, \dots, N, N + 1$) в любой точке $x \in [x_{i-2}; x_{i+2}]$ можно вычислить, используя формулы (5.133) и (5.129).

Для равномерной сетки Δ с постоянным шагом h , учитывая (5.133) и (5.129), получим следующее выражение для нормализованного кубического B -сплайна [28].

$$B_i(x) = \frac{1}{6h^3} \sum_{k=0}^4 (-1)^k C_4^k (x - x_{i+2-k})_+^3, \quad (5.135)$$

где C_4^k — число комбинаций из 4 по k , $i = -1, 0, \dots, N, N + 1$.

По формуле (5.135) можно вычислять значения сплайна $B_i(x)$ в любой точке интервала $[x_{i-2}; x_{i+2}]$ ($i = -1, 0, 1, \dots, N, N + 1$). В табл. 5.5 приведены значения нормализованного

Таблица 5.5

x	$B_i(x)$	$B'_i(x)$	$B''_i(x)$
x_{i-2}	0	0	0
x_{i-1}	$1/6$	$1/2h$	$1/h^2$
x_i	$2/3$	0	$-2/h^2$
x_{i+1}	$1/6$	$-1/2h$	$1/h^2$
x_{i+2}	0	0	0

кубического B -сплайна и его первой и второй производных в узлах интервала-носителя для равномерной сетки Δ с шагом h [18].

Для неравномерной сетки соответствующие значения сплайна $B_i(x)$ и его производных приведены в [18, с. 140]. Там же изложены алгоритмы для вычисления сплайна по рекуррентной формуле (5.97).

Функции $B_i(x)$ ($i = -1, 0, 1, \dots, N, N+1$) линейно независимы на расширенной сетке, их можно выбрать в качестве базиса в пространстве кубических сплайнов дефекта 1. Тогда любой кубический сплайн $S_3(x)$ на отрезке $[a; b]$ можно представить в виде линейной комбинации B -сплайнов

$$S_3(x) = \sum_{i=-1}^{N+1} c_i B_i(x), \\ x \in [a; b], c_i = \text{const.} \quad (5.136)$$

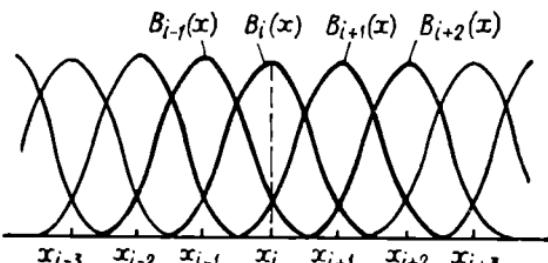


Рис. 17

Выбор B -сплайнов в качестве базисных функций позволяет вычислять значения сплайна $S_3(x)$ и его производных до третьего порядка в любой точке $x \in [a; b]$. При этом, если $x = x_i$, то в формуле (5.136) требуется вычислить значения только трех, а при $x \neq x_i$ — значения четырех базисных сплайнов $B_i(x)$, носитель которых содержит точку x . Например, в узле x_i ненулевыми будут сплайны $B_{i-1}(x)$, $B_i(x)$, $B_{i+1}(x)$, а в точке $x \in (x_i; x_{i+1})$ — сплайны $B_{i-1}(x)$, $B_i(x)$, $B_{i+1}(x)$, $B_{i+2}(x)$ (рис. 17). Поэтому формулу (5.136) можно представить в виде

$$S_3(x) = \sum_{k=t-1}^{t+2} c_k B_k(x), \quad x \in (x_i; x_{i+1}). \quad (5.137)$$

Коэффициенты c_i ($i = -1, 0, 1, \dots, N, N+1$) в представлении (5.136) сплайна $S_3(x)$ определяются из условий интерполяции $S_3(x_i) = y_i$ ($i = 0, 1, \dots, N$) и краевых условий типа (5.123) — (5.126). Каждому узлу x_i ($i = 0, 1, \dots, N$) соответствует уравнение только с тремя неизвестными

$$c_{i-1} B_{i-1}(x_i) + c_i B_i(x_i) + c_{i+1} B_{i+1}(x_i) = y_i, \quad i = 0, 1, \dots, N. \quad (5.138)$$

В случае равномерной сетки Δ с шагом h и краевых условий (5.123) для определения коэффициентов c_i получим

систему уравнений

$$\left\{ \begin{array}{l} \frac{1}{2h} c_{-1} - \frac{1}{2h} c_1 = f'(a); \\ \frac{1}{6} c_{i-1} + \frac{2}{3} c_i + \frac{1}{6} c_{i+1} = y_i, \quad i = 0, 1, \dots, N; \\ -\frac{1}{2h} c_{N-1} + \frac{1}{2h} c_{N+1} = f'(b). \end{array} \right. \quad (5.139)$$

Из первого и последнего уравнений исключим соответственно c_{-1} и c_{N+1} :

$$c_{-1} = c_1 + 2hf'(a), \quad c_{N+1} = c_{N-1} + 2hf'(b).$$

Коэффициенты $c_i (i = 0, 1, \dots, N)$ определяем из системы уравнений

$$\left\{ \begin{array}{l} \frac{2}{3} c_0 + \frac{1}{3} c_1 = y_0 - \frac{1}{3} hf'(a); \\ \frac{1}{6} c_{i-1} + \frac{2}{3} c_i + \frac{1}{6} c_{i+1} = y_i, \quad i = 1, 2, \dots, N-1; \\ \frac{1}{3} c_{N-1} + \frac{2}{3} c_N = y_N - \frac{1}{3} hf'(b). \end{array} \right. \quad (5.140)$$

Матрица системы (5.140) имеет доминирующую главную диагональ. Поэтому существует единственное решение системы, которое может быть найдено методом прогонки. Для краевых условий (5.124) система уравнений для определения коэффициентов c_i имеет вид

$$\left\{ \begin{array}{l} \frac{1}{h^2} c_{-1} - \frac{2}{h^2} c_0 + \frac{1}{h^2} c_1 = f''(a); \\ \frac{1}{6} c_{i-1} + \frac{2}{3} c_i + \frac{1}{6} c_{i+1} = y_i, \quad i = 0, 1, \dots, N; \\ \frac{1}{h^2} c_{N-1} - \frac{2}{h^2} c_N + \frac{1}{h^2} c_{N+1} = f''(b). \end{array} \right. \quad (5.141)$$

Аналогичные системы уравнений получим для периодического сплайна

$$\left\{ \begin{array}{l} c_{-1} = c_0 + c_{N-1} - c_N; \\ \frac{1}{6} c_{i-1} + \frac{2}{3} c_i + \frac{1}{6} c_{i+1} = y_i, \quad i = 1, 2, \dots, N; \\ c_{N+1} = c_1 - c_0 + c_N. \end{array} \right. \quad (5.142)$$

Для краевых условий (5.126)

$$\left\{ \begin{array}{l} c_{-1} = 4c_0 - 6c_1 + 4c_2 - c_3. \\ \frac{1}{6} c_{i-1} + \frac{2}{3} c_i + \frac{1}{6} c_{i+1} = y_i, \quad i = 1, 2, \dots, N-1; \\ c_{N+1} = 4c_N - 6c_{N-1} + 4c_{N-2} - c_{N-3}. \end{array} \right. \quad (5.143)$$

В случае неравномерной сетки Δ матрицы систем (5.140)–(5.143) в общем случае могут не иметь доминирующей главной диагонали.

Во избежание плохой обусловленности матрицы рекомендуется [18] выбирать сетку Δ таким образом, чтобы выполнялось условие

$$\max_{|i-j|=1} \frac{h_i}{h_j} \leq \rho < (3 + \sqrt{5})/2. \quad (5.144)$$

Изложим алгоритм вычисления значений интерполяционного кубического сплайна $S_3(x)$ в любой точке $x \in [a; b]$ с использованием аппарата B -сплайнов в случае равномерной сетки.

1. Для указанного x вычисляем значение $i = [(x - a)/h]$ ($[x]$ — целая часть числа, т. е. наибольшее целое число, не превышающее x) такое, что $x \in [x_i; x_{i+1}]$.

2. В зависимости от заданных краевых условий, составляем систему алгебраических уравнений (5.140)–(5.143) и вычисляем коэффициенты c_i ($i = -1, 0, \dots, N - 1$) методом прогонки.

3. Вычисляем значения сплайнов $B_{i-1}(x)$, $B_i(x)$, $B_{i+1}(x)$ и $B_{i+2}(x)$ в заданной точке x по формуле (5.135).

4. Вычисляем значение сплайна $S_3(x)$ в точке $x \in [x_i; x_{i+1}]$ по формуле (5.137).

Использование аппарата B -сплайнов требует запоминания в процессе счета значений узлов x_i ($i = -3, -2, \dots, N + 3$) и коэффициентов c_i ($i = -1, 0, 1, \dots, N + 1$), т. е. $2(N + 5)$ чисел. По сравнению с кусочно-многочленным представлением здесь достигается экономия памяти машины. Но количество вычислений при этом увеличивается.

В практических вычислениях целесообразно избегать решения системы уравнений для вычисления коэффициентов c_i ($i = -1, 0, 1, \dots, N, N + 1$). Можно воспользоваться приближенными выражениями этих коэффициентов через значения функции $f(x)$ в узлах интерполяции. Например, на равномерной сетке с шагом h можно положить [13]

$$c_i = y_i, \quad i = 0, 1, \dots, N \quad (5.145)$$

с точностью $O(h^2)$. Если же выбрать

$$c_i = -\frac{1}{6}y_{i-1} + \frac{4}{3}y_i - \frac{1}{6}y_{i+1}, \quad i = 0, 1, \dots, N, \quad (5.146)$$

то порядок погрешности будет $O(h^4)$ [18]. Сплайн $S_3(x)$ уже не интерполирует функцию $f(x)$ в узлах сетки Δ , но точность аппроксимации почти не изменяется.

Продифференцировав выражение (5.136), получим

$$S_3^{(r)}(x) = \sum_{i=-1}^{N+1} c_i B_i^{(r)}(x), \quad r = 1, 2, 3, \quad (5.147)$$

т. е. производные кубического сплайна, выраженные через производные B -сплайнов.

Свойство локальности B -сплайнов обеспечивает хорошую гибкость аппарата приближения на их основе. Если сплайн $S_3(x)$ выражен через линейную комбинацию B -сплайнов, то изменение значения функции $f(x)$ в одном из узлов влечет за собой изменение только трех базисных функций, не нарушая непрерывности кривой и ее поведения на некотором удалении от узла. Это позволяет проводить локальную коррекцию формы кривой, например в задачах проектирования, без полного ее пересчета. B -сплайны также мало чувствительны к погрешностям в исходных данных, алгоритмы их вычисления устойчивы относительно погрешностей округления.

Применение явных методов аппроксимации функции $f(x)$ B -сплайнами с использованием формул (5.145), (5.146), не требующих решения систем линейных алгебраических уравнений, существенно уменьшает объем вычислений. Так, при аппроксимации коэффициентов c_i ($i = -1, 0, 1, \dots, N + 1$) по формулам (5.145) в описанном выше алгоритме вычисления сплайна в точке $x \in [x_i; x_{i+1}]$ не требуется выполнять п. 2, а в п. 4 вычисляется

$$S_3(x) = \sum_{k=i-1}^{i+2} y_k B_k(x), \quad x \in (x_i; x_{i+1}). \quad (5.148)$$

Рассмотрим реализацию на микрокалькуляторе алгоритма вычисления значений B -сплайна и интерполяционного сплайна в точке x , принадлежащей внутреннему частичному отрезку $[x_i; x_{i+1}]$ интервала $[a; b]$. Для сетки Δ с постоянным шагом h выпишем выражения для сплайнов $B_k(x)$ ($k = i-1, i, i+1, i+2$) (5.135):

$$B_{i-3}(x) = \frac{1}{6h^3} [-4(x - x_i)^3 + 6(x - x_{i-1})^3 - 4(x - x_{i-2})^3 + (x - x_{i-3})^3];$$

$$B_i(x) = \frac{1}{6h^3} [6(x - x_i)^3 - 4(x - x_{i-1})^3 + (x - x_{i-2})^3];$$

$$B_{i+1}(x) = \frac{1}{6h^3} [-4(x - x_i)^3 + (x - x_{i-1})^3];$$

$$B_{i+2}(x) = \frac{1}{6h^3} (x - x_i)^3.$$

Интервалы-носители каждого из этих B -сплайнов содержат точку x .

Более устойчивый алгоритм вычисления значений нормализованных B -сплайнов в точке $x \in [x_i; x_{i+1}]$ можно получить, применяя рекуррентную формулу (5.97), которая пригодна и для неравномерной сетки. Здесь последовательно строятся сплайны нулевой, первой, второй и третьей степеней $B_{n,i}(x)$ для интервалов-носителей $[x_i; x_{i+n+1}]$ ($n = 0, \dots, 3$). Полагаем

$$B_{0,i}(x) = \begin{cases} 1, & x \in (x_i; x_{i+1}), \\ 0, & x \notin (x_i; x_{i+1}). \end{cases}$$

Затем по (5.97) формируем таблицу:

$$\begin{array}{cccc} B_{0,i}(x) & B_{1,i}(x) & B_{2,i}(x) & B_{3,i}(x) \\ B_{1,i-1}(x) & B_{2,i-1}(x) & B_{3,i-1}(x) & \\ & B_{2,i-2}(x) & B_{3,i-2}(x) & \\ & & B_{3,i-3}(x) & \end{array}$$

Согласно (5.97):

$$\begin{aligned} B_{1,i}(x) &= \frac{x - x_i}{x_{i+3} - x_i} B_{0,i}(x) = \frac{x - x_i}{x_{i+3} - x_i}; \\ B_{1,i-1}(x) &= \frac{x_{i+1} - x}{x_{i+1} - x_i} B_{0,i}(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}; \\ B_{2,i}(x) &= \frac{x - x_i}{x_{i+2} - x_i} B_{1,i}(x); \\ B_{2,i-1}(x) &= \frac{x - x_{i-1}}{x_{i+1} - x_{i-1}} B_{1,i-1}(x) + \frac{x_{i+2} - x}{x_{i+2} - x_i} B_{1,i}(x); \\ B_{2,i-2}(x) &= \frac{x_{i+1} - x}{x_{i+1} - x_{i-1}} B_{2,i-1}(x); \\ B_{3,i}(x) &= \frac{x - x_i}{x_{i+3} - x_i} B_{2,i}(x); \\ B_{3,i-1}(x) &= \frac{x - x_{i-1}}{x_{i+2} - x_{i-1}} B_{2,i-1}(x) + \frac{x_{i+3} - x}{x_{i+3} - x_i} B_{2,i}(x); \\ B_{3,i-2}(x) &= \frac{x - x_{i-2}}{x_{i+1} - x_{i-2}} B_{2,i-2}(x) + \frac{x_{i+2} - x}{x_{i+2} - x_{i-1}} B_{2,i-1}(x); \\ B_{3,i-3}(x) &= \frac{x_{i+1} - x}{x_{i+1} - x_{i-2}} B_{2,i-2}(x). \end{aligned}$$

Построенные таким образом сплайны третьей степени $B_{3,i-3}(x), B_{3,i-2}(x), B_{3,i-1}(x), B_{3,i}(x)$ в случае равномерной сетки совпадают соответственно со сплайнами $B_{i-1}(x), B_i(x), B_{i+1}(x), B_{i+2}(x)$.

Значение интерполяционного кубического сплайна $S_3(x)$ в точке $x \in [x_i; x_{i+1}]$ вычисляется по (5.137) так:

$$S_3(x) = \sum_{k=i-1}^{i+2} c_k B_k(x).$$

Коэффициенты c_k определяются в зависимости от краевых условий по (5.139) — (5.142). Здесь использована локальная аппроксимация при $c_k = y_k$ ($k = 0, 1, \dots, N$).

Программа 28 вычисления значения сплайн-функции $S_3(x)$ в точке $x \in [x_i; x_{i+1}]$ ($i = 1, 2, \dots, N - 2$), выраженной по формуле (5.148) через кубические B -сплайны.

```

СХ П9 ИПА ИПВ — ПА 8 ПО 0 ПД
ИПО П1 ИПА ПВ ↑ FX2 × КИП1 × ИПД
+ ПД ИПВ ИПС + FL1 13 КИП↑ ИПД ×
ИП9 + П9 FL0 35 FL0 08 6 ÷ ИПС
↑ FX2 × ÷ С/П

```

Распределение регистров памяти:

Номер регистра	0,1	3,7	5	8	6	4	2	Д, 9
Переменная	счетчики циклов	$-C_4^1$	C_4^2	C_{i-1}	C_i	C_{i+1}	C_{i+2}	рабочие
	A	B	C					
	$x, x - x_i$	x_i, x_{i+2-k}	h					

Инструкция. 1. Ввести программу в режиме программирования (F ПРГ).

2. В режиме автоматической работы (F АВТ) ввести исходные данные:

(x) ПА (x_i) ПВ (h) ПС (-4) П3 П7 (6) П5
 (c_{i-1}) П8 (c_i) П6 (c_{i+1}) П4 (c_{i+2}) П2

3. Нажать клавиши В/О С/П. После ОСТАНОВа прочесть на индикаторе значение $S_3(x)$.

Контрольный пример. Рассмотрим аппроксимацию с помощью кубических B -сплайнов функции $f(x) = e^x$, заданной на отрезке $[0; 0,5]$ таблицей значений в узлах с шагом $h = 0,1$. Вычислим значение $S_3(0,25)$, используя локальную аппроксимацию, т. е. полагая $c_i = y_i$.

После ввода программы в автоматическом режиме (F АВТ) введем исходные данные: 0,25 ПА 0,2 ПВ 0,1 ПС (-4) ПЗ П7 6 П5 1.10517 П8 1.22140 П6 1.34986 П4 1.49182 П2. Осуществляем пуск программы: В/О С/П. Примерно через 70 с получим результат $S_3(0,25) = 1.286166$. С точным значением функции $e^{0,25} = 1.2840254\dots$ совпадают 2 знака после запятой. Порядок погрешности $O(10^{-2})$ вызван локальной аппроксимацией коэффициентов c_i и округлениями табличных значений функции. По сравнению с аппроксимацией кусочно-полиномиальными сплайнами по программам 26 и 27 погрешность в данном случае больше, но зато здесь не нужно решать систему алгебраических уравнений для определения параметров сплайна.

Глава 6. ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ И ИНТЕГРИРОВАНИЕ ФУНКЦИЙ ОДНОЙ ПЕРЕМЕННОЙ

6.1. Численное дифференцирование функций с помощью интерполяционных полиномов

Численное дифференцирование применяется при решении задач, в которых нужно найти производную некоторого порядка функции, заданной таблично или в виде сложного аналитического выражения. Пусть функция $f(x)$ задана значениями $y_i = f(x_i)$ в $(n+1)$ -й точке x_i ($i = 0, 1, 2, \dots, n$) отрезка $[a; b]$. Общий метод получения формул численного дифференцирования состоит в аппроксимации табличных значений функции каким-либо интерполяционным многочленом $P_n(x)$. Искомые значения производных функции в любой точке отрезка $[a; b]$ приближенно выражаются через соответствующие значения производных этого многочлена:

$$f^{(k)}(x) \approx P_n^{(k)}(x), \quad 0 \leq k \leq n, \quad x \in [a; b]. \quad (6.1)$$

Если известна погрешность интерполяционной формулы

$$|R_n(x)| = |f(x) - P_n(x)|,$$

то погрешности производных

$$|R_n^{(k)}(x)| = |f^{(k)}(x) - P_n^{(k)}(x)|, \quad 0 \leq k \leq n. \quad (6.2)$$

Рассмотрим частные случаи формул численного дифференцирования, в которых функция $f(x)$ приближается интерполяционными полиномами Ньютона, Лагранжа или сплайнами.

1. *Применение интерполяционных формул Ньютона.* Пусть функция $f(x)$, заданная на отрезке $[a; b]$ таблицей значений в равноотстоящих точках x_i ($i = 0, 1, 2, \dots, n$) с шагом

$h = x_{i+1} - x_i$, аппроксимируется интерполяционным полиномом Ньютона для интерполяции вперед (5.15):

$$f(x) \approx y_0 + t\Delta y_0 + \frac{t(t-1)}{2!} \Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!} \Delta^3 y_0 + \\ + \frac{t(t-1)(t-2)(t-3)}{4!} \Delta^4 y_0 + \dots + \frac{t(t-1)\dots(t-n+1)}{n!} \Delta^n y_0, \quad (6.3)$$

где $t = \frac{x - x_0}{h}$.

Последовательно дифференцируя многочлен (6.3) по переменной x и учитывая, что

$$\frac{df(x)}{dx} = \frac{df(x)}{dt} \cdot \frac{dt}{dx} = \frac{df(x)}{dt} \cdot \frac{1}{h},$$

получим формулы для приближенного вычисления производных функции $f(x)$:

$$f'(x) \approx \frac{1}{h} \left[\Delta y_0 + \frac{2t-1}{2} \Delta^2 y_0 + \frac{3t^2-6t+2}{3!} \Delta^3 y_0 + \right. \\ \left. + \frac{4t^3-18t^2+22t-6}{4!} \Delta^4 y_0 + \frac{5t^4-40t^3+105t^2-100t+24}{5!} \Delta^5 y_0 + \dots \right]; \\ f''(x) \approx \frac{1}{h^2} \left[\Delta^2 y_0 + (t-1) \Delta^3 y_0 + \frac{12t^2-36t+22}{4!} \Delta^4 y_0 + \right. \\ \left. + \frac{20t^3-120t^2+210t-100}{5!} \Delta^5 y_0 + \dots \right]; \\ f'''(x) \approx \frac{1}{h^3} \left[\Delta^3 y_0 + \frac{24t-36}{4!} \Delta^4 y_0 + \frac{60t^2-240t+210}{5!} \Delta^5 y_0 + \dots \right]; \\ f^{IV}(x) \approx \frac{1}{h^4} \left[\Delta^4 y_0 + \frac{120t-240}{5!} \Delta^5 y_0 + \dots \right] \text{ и т. д.} \quad (6.4)$$

Формулы (6.4) удобно применять, если значение x расположено ближе к левому концу отрезка $[a; b]$. Для вычисления значения производной в точке, лежащей ближе к правому концу отрезка $[a; b]$, можно воспользоваться интерполяционным полиномом Ньютона для интерполяции назад (5.16):

$$f(x) \approx y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!} \Delta^2 y_{n-2} + \frac{t(t+1)(t+2)}{3!} \Delta^3 y_{n-3} + \\ + \frac{t(t+1)(t+2)(t+3)}{4!} \Delta^4 y_{n-4} + \dots + \frac{t(t+1)\dots(t+n-1)}{n!} \Delta^n y_0, \quad (6.5)$$

где $t = \frac{x - x_n}{h}$.

Дифференцируя (6.5), получим формулы для приближенного вычисления производных функций:

$$f'(x) \approx \frac{1}{h} \left[\Delta y_{n-1} + \frac{2t+1}{2!} \Delta^2 y_{n-2} + \frac{3t^2+6t+2}{3!} \Delta^3 y_{n-3} + \right. \\ \left. + \frac{4t^3+18t^2+22t+6}{4!} \Delta^4 y_{n-4} + \dots \right] \text{ и т. д.} \quad (6.6)$$

Если требуется вычислить производные в узлах интерполяции x_i ($i = 0, 1, \dots, n$), то формулы (6.4) и (6.6) упрощаются. Поскольку любую точку можно принять за начальную, то, положив в (6.4) $x = x_0$ ($t = 0$), получим формулы для вычисления значений производных:

$$f'(x_0) \approx \frac{1}{h} \left[\Delta y_0 - \frac{1}{2} \Delta^2 y_0 + \frac{1}{3} \Delta^3 y_0 - \frac{1}{4} \Delta^4 y_0 + \frac{1}{5} \Delta^5 y_0 + \dots \right]; \\ f''(x_0) \approx \frac{1}{h^2} \left[\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 - \frac{5}{6} \Delta^5 y_0 + \dots \right]; \quad (6.7) \\ f'''(x_0) \approx \frac{1}{h^3} \left[\Delta^3 y_0 - \frac{3}{2} \Delta^4 y_0 + \frac{7}{6} \Delta^5 y_0 - \dots \right]; \\ f^{IV}(x_0) \approx \frac{1}{h^4} \left[\Delta^4 y_0 - 2\Delta^5 y_0 + \dots \right] \text{ и т. д.}$$

Аналогично, положив в формуле (6.6) $x = x_n$ ($t = 0$), получим

$$f'(x_n) = \frac{1}{h} \left[\Delta y_{n-1} + \frac{1}{2} \Delta^2 y_{n-2} + \frac{1}{3} \Delta^3 y_{n-3} + \frac{1}{4} \Delta^4 y_{n-4} + \dots \right] \\ \text{и т. д.} \quad (6.8)$$

Если в интерполяционном полиноме Ньютона (6.3) ограничиться разностями до k -го порядка, то его остаточный член

$$R_k(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_k)}{(k+1)!} f^{(k+1)}(\xi) = \\ = h^{k+1} \frac{t(t-1)\cdots(t-k)}{(k+1)!} f^{(k+1)}(\xi), \quad (6.9)$$

где ξ — некоторое число, лежащее между узлами интерполяции x_0, x_1, \dots, x_k и x .

Продифференцировав (6.9) по x и положив $x = x_0$, т. е. $t = 0$, получим погрешность для первой производной функции

$$|R_k(x_0)| = \left| (-1)^k \frac{h^k}{(k+1)!} f^{(k+1)}(\xi) \right|. \quad (6.10)$$

На практике оценка (6.10) мало пригодна, так как обычно не известна производная $f^{(k+1)}(x)$. Поэтому при небольших k

можно приближенно заменить производную ее выражением через конечные разности и представить погрешность в виде

$$|R_k(x_0)| = \left| \frac{(-1)^k}{h} \frac{\Delta^{k+1} y_0}{k+1} \right|. \quad (6.11)$$

Аналогично получаются оценки погрешности для производных высших порядков.

Следует учесть, что минимальное число узлов, необходимое для вычисления k -й производной, равно $k + 1$. При использовании большего числа узлов получаются формулы более высокого порядка точности.

Пример. Пусть функция $f(x) = e^{-x^2}$ задана таблицей значений с шагом $h = 0,1$:

x_i	0	0,1	0,2	0,3	0,4	0,5
y_i	1	0,9900	0,9608	0,9139	0,8524	0,7788

Вычислим приближенно значения производных $f'(0)$ и $f''(0)$. Составим таблицу конечных разностей:

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
0	1	—0,0100				
0,1	0,9900	—0,0292	—0,0192	0,0025		
0,2	0,9608	—0,0469	—0,0167	0,0021	—0,0004	
0,3	0,9139	—0,0615	—0,0146	0,0025	0,0004	0,0008
0,4	0,8524	—0,0736	—0,0121			
0,5	0,7788					

Воспользуемся формулой (6.7), полагая $x_0 = 0$ и ограничиваясь разностями четвертого порядка,

$$\begin{aligned} f'(0) &\approx \frac{1}{h} \left(\Delta y_0 - \frac{1}{2} \Delta^2 y_0 + \frac{1}{3} \Delta^3 y_0 - \frac{1}{4} \Delta^4 y_0 \right) = \\ &= \frac{1}{0,1} \left(-0,0100 + \frac{1}{2} \cdot 0,0192 + \frac{1}{3} \cdot 0,0025 + \frac{1}{4} \cdot 0,0004 \right) = 0,0083, \end{aligned}$$

$$\begin{aligned} f''(0) &\approx \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 \right) = \\ &= \frac{1}{0,01} \left(-0,0192 - 0,0025 + \frac{11}{12} \cdot 0,0004 \right) = -2,2067. \end{aligned}$$

Точные значения производных равны соответственно

$$f'(0) = -2xe^{-x^2}|_{x=0} = 0; \quad f''(0) = (4x^2 - 2)e^{-x^2}|_{x=0} = -2.$$

Выразим погрешность через конечные разности пятого порядка по (6.7):

$$|R'_4(0)| \approx \frac{1}{5h} |\Delta^5 y_0| = \frac{1}{5 \cdot 0,1} \cdot 0,0008 = 0,0016,$$

$$|R''_4(0)| \approx \frac{5}{6} \cdot \frac{1}{h^2} |\Delta^5 y_0| = \frac{5}{6 \cdot (0,1)^2} \cdot 0,0008 = 0,0667.$$

Следует отметить, что с увеличением порядка производной точность численного дифференцирования резко падает. На практике их применяют для вычисления производных не выше второго порядка.

2. Применение интерполяционного полинома Лагранжа. Функцию $f(x)$, заданную на отрезке $[a; b]$ значениями $y_i = f(x_i)$ в равноотстоящих точках x_i ($i = 0, 1, \dots, n$; $h = (x_{i+1} - x_i)$, заменим приближенно интерполяционным полиномом Лагранжа (см. (5.10)):

$$f(x) \approx \sum_{i=0}^n \frac{(-1)^{n-i}}{i! (n-i)!} \frac{t(t-1)\cdots(t-n)}{t-i} y_i, \quad (6.12)$$

где $t = \frac{x - x_0}{h}$.

Продифференцируем (6.12) по x :

$$f'(x) \approx \frac{1}{h} \sum_{i=0}^n \frac{(-1)^{n-i}}{i! (n-i)!} \frac{dy_i}{dt} \left[\frac{t(t-1)\cdots(t-n)}{t-i} \right]. \quad (6.13)$$

Учитывая выражение для погрешности интерполяционной формулы Лагранжа

$$|R_n(x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_n(x) \right|, \quad (6.14)$$

после дифференцирования получим оценку погрешности для производной функции в узлах интерполяции

$$|R'_n(x_i)| = \left| (-1)^{n-i} h^n \frac{i! (n-i)!}{(n+1)!} f^{(n+1)}(\xi) \right|, \quad (6.15)$$

где $\xi \in [a; b]$.

В формуле (6.13) производная функции выражается не через конечные разности, а через ее значения в узлах интерполяции. Выпишем некоторые ее частные случаи, позволяющие вычислять производные функции и их погрешности в узлах при фиксированном n .

а. При $n = 2$ (заданы значения функции в трех точках) получим значения первых производных:

$$\begin{aligned} f'(x_0) &= \frac{1}{2h}(-3y_0 + 4y_1 - y_2) + \frac{h^2}{3}f'''(\xi), \\ f'(x_1) &= \frac{1}{2h}(y_2 - y_0) - \frac{h^2}{6}f'''(\xi), \\ f'(x_2) &= \frac{1}{2h}(y_0 - 4y_1 + 3y_2) + \frac{h^2}{3}f'''(\xi), \end{aligned} \quad (6.16)$$

и вторых производных:

$$\begin{aligned} f''(x_0) &= \frac{1}{h^2}(y_0 - 2y_1 + y_2) - hf'''(\xi), \\ f''(x_1) &= \frac{1}{h^2}(y_0 - 2y_1 + y_2) - \frac{h^2}{12}f^{IV}(\xi), \\ f''(x_2) &= \frac{1}{h^2}(y_0 - 2y_1 + y_2) + hf'''(\xi). \end{aligned} \quad (6.17)$$

б. При $n = 3$ (четыре узла интерполяции) получим выражения для первых производных:

$$\begin{aligned} f'(x_0) &= \frac{1}{6h}(-11y_0 + 18y_1 - 9y_2 + 2y_3) - \frac{h^3}{4}f^{IV}(\xi), \\ f'(x_1) &= \frac{1}{6h}(-2y_0 - 3y_1 + 6y_2 - y_3) + \frac{h^3}{12}f^{IV}(\xi), \\ f'(x_2) &= \frac{1}{6h}(y_0 - 6y_1 + 3y_2 + 2y_3) - \frac{h^3}{12}f^{IV}(\xi), \\ f'(x_3) &= \frac{1}{6h}(-2y_0 + 9y_1 - 18y_2 + 11y_3) + \frac{h^3}{4}f^{IV}(\xi), \end{aligned} \quad (6.18)$$

и вторых производных:

$$\begin{aligned} f''(x_0) &= \frac{1}{h^2}(2y_0 - 5y_1 + 4y_2 - y_3) + \frac{11}{12}h^2f^{IV}(\xi), \\ f''(x_1) &= \frac{1}{h^2}(y_0 - 2y_1 + y_2) - \frac{h^2}{12}f^{IV}(\xi), \\ f''(x_2) &= \frac{1}{h^2}(y_1 - 2y_2 + y_3) - \frac{h^2}{12}f^{IV}(\xi), \\ f''(x_3) &= \frac{1}{h^2}(-y_0 + 4y_1 - 5y_2 + 2y_3) + \frac{11}{12}h^2f^{IV}(\xi). \end{aligned} \quad (6.19)$$

На практике наиболее применимы следующие формулы численного дифференцирования для любого внутреннего узла x_i :

$$f'(x_i) = \frac{y_{i+1} - y_i}{h} - \frac{h}{2}f''(\xi), \quad (6.20)$$

$$f'(x_i) = \frac{y_{i+1} - y_{i-1}}{2h} - \frac{h^2}{6}f'''(\xi), \quad (6.21)$$

$$f''(x_i) = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} - \frac{h^2}{12} f^{IV}(\xi), \quad (6.22)$$

которые могут быть получены из (6.14), (6.16), (6.17) и (6.19).

Отметим, что с уменьшением шага h для достаточно гладкой функции величина остаточного члена в формулах дифференцирования, т. е. погрешность метода, уменьшается. При этом значения производных более точны в узлах, расположенных посередине равномерной сетки. Вычислительная погрешность, вызванная неточным заданием значений функции y_i , а также ошибками, возникающими при вычислении конечных разностей, увеличивается при уменьшении h и росте порядка производной. Как видно из (6.20)—(6.22), при малых h значения функции в соседних узлах почти равны, поэтому при вычислении их разности теряются значащие цифры. Деление на h^k , если h мало, может привести к большой абсолютной погрешности в значениях k -й производной. Таким образом, задача аппроксимации производных является по своей природе неустойчивой.

Если функция задана в узлах с абсолютной погрешностью ϵ , то для конкретной формулы численного дифференцирования можно примерно указать оптимальный шаг. Например, определим ошибку метода для (6.21)

$$|R'(x_i)| = \frac{h^2}{6} M_3, \text{ где } M_3 = \max_{x_{i-1} < x < x_{i+1}} |f'''(x)|$$

и ошибку, вызванную погрешностью исходных данных $\frac{2\epsilon}{2h} = \frac{\epsilon}{h}$.

Суммарная погрешность (без учета ошибок округления)

$$\frac{h^2}{6} M_3 + \frac{\epsilon}{h} \quad (6.23)$$

принимает минимальное значение при $h = \sqrt[3]{\frac{3\epsilon}{M_3}}$.

Проведя аналогичные рассуждения для (6.22), получим следующее выражение для оптимального шага:

$$h = 2 \sqrt[4]{\frac{2\epsilon}{M_4}}, \quad (6.24)$$

где

$$M_4 = \max_{x_{i-1} < x < x_{i+1}} |f^{IV}(x)|.$$

Формулы (6.23) и (6.24) можно применять в том случае, если известны оценки соответствующих производных.

Оценку погрешности формулы численного дифференцирования можно проводить по правилу Рунге, если остаточный член формулы имеет структуру

$$R_{(k)}^{(k)} = f^{(k)}(x) - P_n^{(k)}(x, h) = \varphi(x) h^p + O(h^{p+1}), \quad k = 1, 2, \dots \quad (6.25)$$

Вычислим производную по указанной формуле на равномерной сетке узлов сначала с шагом h , а затем с шагом rh .

Учитывая, что

$$\begin{aligned} f^{(k)}(x) - P_n^{(k)}(x, rh) &= \varphi(x) (rh)^p + O((rh)^{p+1}), \\ O(h^{p+1}) &\approx O((rh)^{p+1}), \end{aligned}$$

получим следующую оценку погрешности

$$R^{(k)}(x) \approx \varphi(x) h^p = \frac{P_n^{(k)}(x, h) - P_n^{(k)}(x, rh)}{r^p - 1} + O(h^{p+1}). \quad (6.26)$$

Тогда исходную формулу численного дифференцирования p -го порядка можно уточнить, полагая

$$f^{(k)}(x) \approx P_n^{(k)}(x, h) + \frac{P_n^{(k)}(x, h) - P_n^{(k)}(x, rh)}{r^p - 1}, \quad k = 1, 2, \dots, \quad (6.27)$$

6.2. Численное дифференцирование функций с помощью сплайнов

Как уже отмечалось, интерполяционные сплайны обеспечивают хорошее приближение не только функции $f(x)$, заданной на сетке Δ , но и ее производных. Для вычисления производных достаточно продифференцировать интерполяционный сплайн соответствующее число раз.

Для интерполяционных кубических сплайнов формулы численного дифференцирования, выраженные через наклоны $m_i = S'(x_i)$ и моменты $M_i = S''(x_i)$, имеют вид:

$$\begin{aligned} S_3'(x) &= 6t(1-t) \frac{y_{i+1} - y_i}{h_i} + (1 - 4t + 3t^2)m_i - \\ &\quad - (2t - 3t^2)m_{i+1}; \end{aligned} \quad (6.28)$$

$$S_3''(x) = (6 - 12t) \frac{y_{i+1} - y_i}{h_i^2} - \frac{4 - 6t}{h_i} m_i - \frac{2 - 6t}{h_i} m_{i+1}; \quad (6.29)$$

$$S_3'''(x) = \frac{6}{h_i^2} \left(m_{i+1} + m_i - 2 \frac{y_{i+1} - y_i}{h_i} \right); \quad (6.30)$$

$$S'_3(x) = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6} (2M_i + M_{i+1}) + M_i(x - x_i) + \\ + \frac{M_{i+1} - M_i}{2h_i} (x - x_i)^2; \quad (6.31)$$

$$S''_3(x) = M_i + \frac{M_{i+1} - M_i}{h_i} (x - x_i); \quad (6.32)$$

$$S'''_3(x) = \frac{M_{i+1} - M_i}{h_i}, \quad (6.33)$$

где $x \in [x_i; x_{i+1}]$, $h_i = x_{i+1} - x_i$ ($i = 0, 1, \dots, N-1$), $t = \frac{x - x_i}{h_i}$.

Если $f(x) \in C^4[a; b]$, то порядок приближения производных $f'(x)$, $f''(x)$ и $f'''(x)$ сплайнами $S'(x)$, $S''(x)$, $S'''(x)$ равен соответственно $O(h^3)$, $O(h^2)$, $O(h)$.

Если в качестве базиса взяты кубические B -сплайны, то производные аппроксимирующей функции выражаются через производные B -сплайнов:

$$S_3^{(r)}(x) = \sum_{i=-1}^{N+1} c_i B_i^{(r)}(x), \quad r = 1, 2, 3. \quad (6.34)$$

Алгоритмы вычисления m_i , M_i и c_i изложены в гл. 5. Если воспользоваться локальной аппроксимацией функции и положить $c_i = y_i$ ($i = 0, 1, \dots, N$), то производные сплайна

$$S_3^{(r)}(x) = \sum_{i=-1}^{N+1} y_i B_i^{(r)}(x), \quad r = 1, 2, 3. \quad (6.35)$$

Значения производных функции $f(x)$ в узлах интерполяции x_i ($i = 0, 1, \dots, N$) приближенно приравниваем к значениям соответствующих производных кубического сплайна. Получим следующие формулы численного дифференцирования в узлах:

$$S'_3(x_i) = m_i, \quad i = 0, 1, \dots, N; \quad (6.36)$$

$$S'_3(x_i) = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6} (2M_i + M_{i+1}), \quad i = 0, 1, \dots, N-1; \quad (6.37)$$

$$S'_3(x_i) = \frac{y_i - y_{i-1}}{h_i} + \frac{h_i}{6} (M_{i-1} + 2M_i), \quad i = 1, 2, \dots, N; \quad (6.38)$$

$$S''_3(x_i) = M_i, \quad i = 0, 1, \dots, N; \quad (6.39)$$

$$S''_3(x_i) = 6 \frac{y_{i+1} - y_i}{h_i^2} - \frac{2}{h_i} (2m_i + m_{i+1}), \quad i = 0, 1, \dots, N-1; \quad (6.40)$$

$$S_3^{'}(x_i) = \frac{2}{h_{i-1}}(m_{i-1} + 2m_i) - 6\frac{y_i - y_{i-1}}{h_{i-1}^2}, \quad i = 1, 2, \dots, N; \quad (6.41)$$

$$S_3^{''}(x_i) = \frac{6}{h_i^2} \left(m_{i+1} + m_i - 2\frac{y_{i+1} - y_i}{h_i} \right), \quad i = 0, 1, \dots, N-1; \quad (6.42)$$

$$S_3^{'''}(x_i) = \frac{M_{i+1} - M_i}{h_i}, \quad i = 0, 1, \dots, N-1; \quad (6.43)$$

$$S_3^{(r)}(x_i) = \sum_{k=i-1}^{i+1} c_k B_k^{(r)}(x_i), \quad r = 1, 2, 3; \quad i = 0, 1, \dots, N. \quad (6.44)$$

Если выбрать сетку Δ равномерной с шагом h , а число узлов интерполяции достаточно большим, то внутри промежутка $[a; b]$ точность аппроксимации производных функции повышается. Для равномерной сетки значения производных нормализованных кубических B -сплайнов приведены в табл. 5.5. Учитывая (6.44), получим

$$S_3^{'}(x_i) = c_{i-1} B_{i-1}^{'}(x_i) + c_i B_i^{'}(x_i) + c_{i+1} B_{i+1}^{'}(x_i) = \frac{c_{i+1} - c_{i-1}}{2h}, \quad (6.45)$$

$$i = 0, 1, \dots, N.$$

$$S_3^{''}(x_i) = c_{i-1} B_{i-1}^{''}(x_i) + c_i B_i^{''}(x_i) + c_{i+1} B_{i+1}^{''}(x_i) =$$

$$= \frac{c_{i-1} - 2c_i + c_{i+1}}{h^2}, \quad i = 0, 1, \dots, N. \quad (6.46)$$

Комбинируя сплайновую и разностную аппроксимации, можно получить формулы численного дифференцирования более высокого порядка точности [18]:

$$f''(x_i) = \frac{1}{2} \left(M_i + \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} \right) + O(h^4); \quad (6.47)$$

$$f'''(x_i) = \frac{1}{12} (M_{i-1} + 10M_i + M_{i+1}) + O(h^4); \quad (6.48)$$

$$f''''(x_i) = \frac{1}{2} \left(\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + 6\frac{y_i - c_i}{h^2} \right) + O(h^4); \quad (6.49)$$

$$f^{IV}(x_i) = \frac{M_{i+1} - M_{i-1}}{2h} + O(h^2); \quad (6.50)$$

$$f^{IV}(x_i) = \frac{M_{i-1} - 2M_i + M_{i+1}}{h^2} + O(h^4); \quad i = 1, 2, \dots, N-1. \quad (6.51)$$

Если значения функции $f(x)$ заданы в узлах с абсолютной погрешностью ϵ на густой сетке, то для численного дифференцирования строится интерполяционный сплайн на более редкой сетке узлов. Оптимальный по порядку шаг сетки выби-

рается $h = O(\epsilon^{\frac{1}{4}})$, что согласуется с формулой (6.24) для полиномиальной интерполяции [28].

Вычисление второй производной функции $f(x)$ можно осуществить дифференцированием сплайна от сплайна [1]. При этом вначале вычисляются наклоны $m_i = S'(x_i)$ по заданным значениям функции, а затем строится сплайн, интерполирующий в узлах значения m_i . Производная построенного сплайна приближает вторую производную функции.

6.3. Численное интегрирование функций одной переменной

При решении многих инженерных задач приходится вычислять определенные интегралы от функций. Из курса математического анализа известно, что для всякой непрерывной на отрезке $[a; b]$ функции $f(x)$ существует первообразная $F(x)$, и определенный интеграл от этой функции можно вычислить по формуле Ньютона—Лейбница

$$\int_a^b f(x) dx = F(b) - F(a).$$

Однако первообразная $F(x)$ не всегда является элементарной функцией. В приложениях приходится также вычислять определенные интегралы, если подынтегральная функция $f(x)$ не является элементарной, либо задается таблично или графически. Тогда формула Ньютона—Лейбница неприменима. Кроме того, выражение для $F(x)$ может оказаться достаточно сложным, требующим громоздких вычислений. В таких случаях применяют формулы приближенного интегрирования, называемые квадратурными.

Идея численных методов интегрирования заключается в замене подынтегральной функции $f(x)$ некоторой аппроксимирующей функцией $\varphi(x)$, интеграл от которой вычисляется легко. В частности, если заменить функцию $f(x)$ на отрезке $[a; b]$ интерполяционным многочленом $P_k(x)$ с узлами интерполяции $x_i \in [a; b]$, то квадратурная формула примет вид:

$$\int_a^b f(x) dx \approx \sum_{i=0}^k A_i f(x_i). \quad (6.52)$$

Числа A_i называют весами, а x_i — узлами квадратурной формулы. Веса A_i не зависят от функции $f(x)$, а зависят только от выбора узлов. Если узлы равноотстоящие с шагом

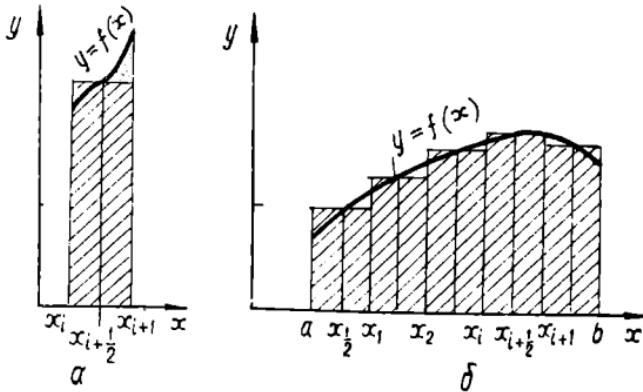


Рис. 18

$h = \frac{b-a}{k}$, то квадратурные формулы (6.52) называют формулами Ньютона—Котеса при разных k .

Для повышения точности квадратурной формулы отрезок интегрирования разбивают на ряд частичных интервалов, на каждом из которых подынтегральную функцию заменяют интерполяционным многочленом. Такую квадратурную формулу называют составной.

Если концы отрезка $[a; b]$ и частичных интервалов интегрирования выбираются в качестве узлов интерполяции, то формулу называют квадратурной формулой замкнутого типа, в противном случае — открытого типа.

Квадратурная формула называется точной для многочленов степени k , если при замене функции $f(x)$ произвольным многочленом $P_k(x)$ степени k приближенное равенство (6.52) становится точным.

Рассмотрим наиболее часто применяемые квадратурные формулы, являющиеся частными случаями формул Ньютона—Котеса при $k = 0, 1, 2$.

Формула прямоугольников. Выберем произвольно отрезок $[x_i; x_{i+1}] \subset [a; b]$ длины $h = x_{i+1} - x_i$ и обозначим средину этого отрезка $x_i + h/2$ через $x_{i+\frac{1}{2}}$. В формуле прямоугольников определенный интеграл от функции $f(x)$ на отрезке $[x_i; x_{i+1}]$ приближенно выражается через значение функции в средней точке $x_{i+\frac{1}{2}}$ отрезка интегрирования и его

длину h :

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx f(x_{i+\frac{1}{2}}) h. \quad (6.53)$$

В данном случае $f(x)$ аппроксимируется многочленом нулевой степени ($k = 0$). Если $f(x) > 0$ при $x \in [x_i; x_{i+1}]$, то формула (6.53) приближенно выражает площадь криволинейной трапеции, ограниченной графиком функции $f(x)$ на отрезке $[x_i; x_{i+1}]$, через площадь прямоугольника, заштрихованного на рис. 18, а.

Оценим погрешность формулы прямоугольников (6.53). Предположим, что функция $f(x)$ имеет на отрезке $[a; b]$ непрерывную вторую производную. Запишем для нее формулу Тейлора в окрестности точки $x_{i+\frac{1}{2}}$ с остаточным членом в форме

Лагранжа, ограничиваясь тремя слагаемыми:

$$f(x) = f\left(x_{i+\frac{1}{2}}\right) + f'\left(x_{i+\frac{1}{2}}\right)(x - x_{i+\frac{1}{2}}) + \frac{f''(\xi_i)}{2!}(x - x_{i+\frac{1}{2}})^2, \quad (6.54)$$

где $\xi_i \in [x_i; x_{i+1}]$.

Тогда

$$\begin{aligned} \int_{x_i}^{x_{i+1}} f(x) dx &= \int_{x_i}^{x_{i+1}} \left(f\left(x_{i+\frac{1}{2}}\right) + f'\left(x_{i+\frac{1}{2}}\right)(x - x_{i+\frac{1}{2}}) + \right. \\ &\quad \left. + \frac{f''(\xi_i)}{2!}(x - x_{i+\frac{1}{2}})^2 \right) dx = f\left(x_{i+\frac{1}{2}}\right) h + \left[f'\left(x_{i+\frac{1}{2}}\right) \times \right. \\ &\quad \times \left. \left. \frac{\left(x - x_{i+\frac{1}{2}}\right)^2}{2} + \frac{f''(\xi_i)}{2!} \frac{\left(x - x_{i+\frac{1}{2}}\right)^3}{3} \right] \right|_{x_i}^{x_{i+1}} = \\ &= f\left(x_{i+\frac{1}{2}}\right) h + \frac{h^3}{24} f''(\xi_i). \end{aligned} \quad (6.55)$$

Учитывая (6.55) и (6.53), получим выражение для остаточного члена формулы прямоугольников

$$R_i = \int_{x_i}^{x_{i+1}} f(x) dx - f\left(x_{i+\frac{1}{2}}\right) h = \frac{h^3}{24} f''(\xi_i), \quad \xi_i \in [x_i; x_{i+1}]. \quad (6.56)$$

Таким образом, формула (6.53) будет тем точнее, чем меньше h .

Для вычисления интеграла $\int_a^b f(x) dx$ разобьем отрезок $[a; b]$ на n равных частей точками $x_0 = a < x_1 < \dots < x_n = b$

с шагом $h = (b - a)/n$. Применяя свойство аддитивности определенного интеграла относительно промежутка интегрирования

$$\int_a^b f(x) dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx$$

и формулу прямоугольников (6.53) для каждого частичного отрезка, получим составную формулу прямоугольников (или формулу средних):

$$\int_a^b f(x) dx = h \left(f\left(x_{0+\frac{1}{2}}\right) + f\left(x_{1+\frac{1}{2}}\right) + \cdots + f\left(x_{n-1+\frac{1}{2}}\right) \right). \quad (6.57)$$

Остаточный член составной формулы прямоугольников (6.57)

$$\begin{aligned} R &= \sum_{i=0}^{n-1} R_i = \frac{h^3}{24} [f''(\xi_0) + f''(\xi_1) + \cdots + f''(\xi_{n-1})] = \\ &= \frac{(b-a)^3}{24n^2} \sum_{i=0}^{n-1} \frac{f''(\xi_i)}{n} = h^2 \frac{b-a}{24} f''(\xi), \end{aligned} \quad (6.58)$$

где $\xi \in [a; b]$.

Была использована формула (6.56) и формула усреднения для непрерывной функции $f''(x)$.

На рис. 18,б дана геометрическая интерпретация составной формулы прямоугольников: площадь криволинейной трапеции, ограниченной графиком функции $f(x)$ на отрезке $[a; b]$, приближенно заменяется суммой площадей заштрихованных прямоугольников.

Формула прямоугольников (6.57) является квадратурной формулой открытого типа. Она точна для многочленов первой степени (их вторая производная равна нулю на $[a; b]$). Погрешность для дважды дифференцируемой на $[a; b]$ функции $f(x)$ равна $O(h^2)$.

Формула трапеций. Аппроксимируем функцию $f(x)$ на отрезке $[x_i; x_{i+1}]$ длины $h = x_{i+1} - x_i$ многочленом первой степени ($k = 1$). Получим формулу трапеций

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx h \frac{f(x_i) + f(x_{i+1})}{2}. \quad (6.59)$$

Геометрическая интерпретация формулы (6.59) при $f(x) > 0$ для $x \in [x_i; x_{i+1}]$ дана на рис. 19,а: криволинейная трапеция приближенно заменяется прямолинейной трапецией.

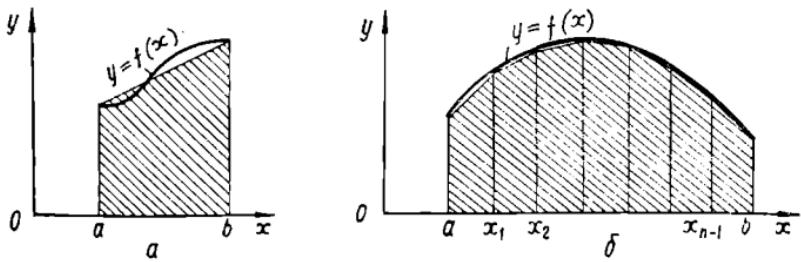


Рис. 19

Найдем выражение для остаточного члена формулы трапеций (6.59), предполагая непрерывность $f''(x)$ на отрезке $[a; b]$.

Подынтегральная функция $f(x)$ на каждом отрезке $[x_i; x_{i+1}]$ заменяется интерполяционным полиномом первой степени $P_1(x)$. Остаточный член интерполяционной формулы Ньютона (5.15) k -го порядка имеет вид (6.9). Полагая в (6.9) $k = 1$, вычислим остаточный член формулы трапеций (6.59) на отрезке $[x_i; x_{i+1}]$:

$$R_i = \int_{x_i}^{x_{i+1}} R_1(x) dx = h^2 \int_0^1 \frac{t(t-1)}{2!} f''(\xi_i) dt = -\frac{h^3}{12} f''(\xi_i),$$

$$\xi_i \in [x_i; x_{i+1}], \quad i = 0, 1, \dots, n-1.$$

Окончательное выражение для остаточного члена формулы трапеций

$$R_i = \int_{x_i}^{x_{i+1}} f(x) dx - h \frac{f(x_i) + f(x_{i+1})}{2} = -\frac{h^3}{12} f''(\xi_i),$$

$$\xi_i \in [x_i; x_{i+1}], \quad i = 0, 1, \dots, n-1. \quad (6.60)$$

Просуммировав значения (6.59) по всем частичным отрезкам, получим составную формулу трапеций для интеграла от функции $f(x)$ на отрезке $[a; b] = \bigcup_{i=0}^{n-1} [x_i; x_{i+1}]$, где $x_i = a + ih$ ($i = 0, 1, \dots, n-1$):

$$\int_a^b f(x) dx \approx h \left(\frac{f(a) + f(b)}{2} + f(x_1) + f(x_2) + \dots + f(x_{n-1}) \right). \quad (6.61)$$

Найдем выражение для остаточного члена составной формулы трапеций (6.61):

$$R = \sum_{i=0}^{n-1} R_i = -\frac{h^3}{12} \sum_{i=0}^{n-1} f''(\xi_i) = -\frac{(b-a)^3}{12n^2} \sum_{i=0}^{n-1} \frac{f''(\xi_i)}{n} = -\frac{h^2(b-a)}{12} f''(\xi), \quad (6.62)$$

где $\xi \in [a; b]$.

Формула трапеций (6.61) является квадратурной формулой замкнутого типа, она точна для многочленов первой степени.

Погрешность для дважды дифференцируемой на $[a; b]$ функции $f(x)$ равна $O(h^2)$. Геометрическая интерпретация составной формулы трапеций приведена на рис. 19,б.

Формула Симпсона. Разобьем интервал интегрирования $[a; b]$ на четное число равных частей с шагом $h = (b-a)/2n$. На каждом отрезке $[x_i; x_{i+2}]$ длины $2h$ заменим функцию $f(x)$ функцией квадратной параболы $\varphi(x) = Ax^2 + Bx + C$ ($k = 2$), интерполирующей $f(x)$ в узлах x_i, x_{i+1}, x_{i+2} .

Вычислим вначале площадь, ограниченную снизу отрезком $[-h; h]$ оси Ox , прямыми $x = -h, x = h$ и параболой $y = Ax^2 + Bx + C$, учитывая, что $f(x_0) = Ah^2 - Bh + C$, $f(x_1) = C$, $f(x_2) = Ah^2 + Bh + C$.

Получим

$$\int_{-h}^h (Ax^2 + Bx + C) dx = \left(A \frac{x^3}{3} + B \frac{x^2}{2} + Cx \right) \Big|_{-h}^h = \frac{2Ah^3}{3} + 2Ch = \frac{h}{3} (2Ah^2 + 6C) = \frac{h}{3} (f(x_0) + 4f(x_1) + f(x_2)). \quad (6.63)$$

Квадратурная формула Симпсона для отрезка $[x_i; x_{i+2}]$ имеет вид:

$$\int_{x_i}^{x_{i+2}} f(x) dx \approx \int_{x_i}^{x_{i+2}} (Ax^2 + Bx + C) dx = \frac{h}{3} (f(x_i) + 4f(x_{i+1}) + f(x_{i+2})). \quad (6.64)$$

Геометрическая интерпретация формулы (6.64) дана на рис. 20,а.

Найдем выражение для остаточного члена формулы Симпсона: Предположим, что функция $f(x)$ имеет на отрезке $[x_i; x_{i+2}]$ непрерывные производные до четвертого порядка.

Формула Симпсона точна для всякого многочлена третьей степени. Поэтому полагаем в остаточном члене интерполя-

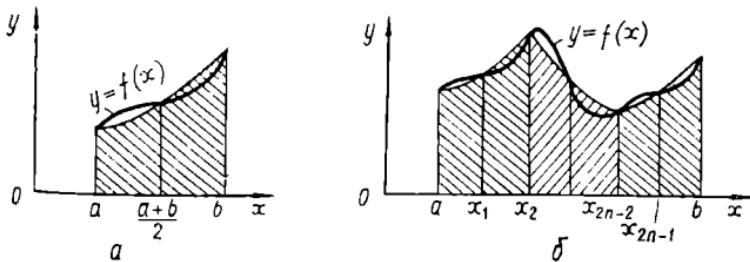


Рис. 20

ционной формулы Ньютона (6.9) $k = 3$ и проинтегрируем его на отрезке $[x_i; x_{i+2}]$ длины $2h$:

$$R_i = \int_{x_i}^{x_{i+2}} R_3(x) dx = h^4 \int_0^2 \frac{t(t-1)(t-2)(t-3)}{4!} f^{IV}(\xi_i) dt = -\frac{h^5}{90} f^{IV}(\xi_i).$$

Тогда разность

$$R_i = \int_{x_i}^{x_{i+2}} f(x) dx - \frac{h}{3}(f(x_i) + 4f(x_{i+1}) + f(x_{i+2})) = -\frac{h^5}{90} f^{IV}(\xi_i), \quad \xi_i \in [x_i; x_{i+2}], \quad i = 0, 2, \dots, 2n-2 \quad (6.65)$$

будет остаточным членом квадратурной формулы Симпсона (6.64) на отрезке $[x_i; x_{i+2}]$.

Просуммировав значения (6.64) для интегралов по всем частичным отрезкам ($i = 0, 2, \dots, 2n-2$), получим составную квадратурную формулу Симпсона (или формулу парабол):

$$\int_a^b f(x) dx \approx \frac{h}{3} [f(a) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \\ + 2f(x_4) + \dots + 2f(x_{2n-2}) + 4f(x_{2n-1}) + f(b)]. \quad (6.66)$$

Геометрическая иллюстрация формулы (6.66) дана на рис. 20, б. Остаточный член составной квадратурной формулы Симпсона (6.66) на отрезке $[a; b]$ с учетом (6.65)

$$R = \sum_{i=0}^{n-1} R_i = -\frac{h^5}{90} \sum_{i=0}^{n-1} f^{IV}(\xi_i) = -\frac{(b-a)^5}{180(2n)^4} \sum_{i=0}^{n-1} \frac{f^{IV}(\xi_i)}{n} = -\frac{h^4(b-a)}{180} f^{IV}(\xi), \quad \xi \in [a; b]. \quad (6.67)$$

Формула Симпсона (6.66) является квадратурной формулой замкнутого типа, она точна для многочленов третьей степени (их четвертая производная равна нулю на $[a; b]$), погрешность для функции $f(x)$, имеющей на $[a; b]$ непрерывную производную $f^{IV}(x)$, равна $O(h^4)$.

Сходимость квадратурных формул. Оценка погрешности по правилу Рунге. Формула прямоугольников (6.57) приближенно выражает определенный интеграл через интегральную сумму при равномерном разбиении отрезка $[a; b]$ на части с шагом h . Формула трапеций (6.61) тоже является интегральной суммой для функции $f(x)$, если отрезок $[a; b]$ разбить на части:

$$\left[x_0; x_0 + \frac{h}{2} \right], \left[x_0 + \frac{h}{2}; x_1 + \frac{h}{2} \right], \dots, \left[x_i - \frac{h}{2}, x_i + \frac{h}{2} \right], \dots, \\ \left[x_n - \frac{h}{2}, x_n \right].$$

Формула Симпсона (6.66) является линейной комбинацией формул прямоугольников и трапеций:

$$I_3 = \frac{2}{3} I_1 + \frac{1}{3} I_2,$$

где I_1 , I_2 и I_3 — приближенные значения интеграла $\int_a^b f(x) dx$, полученные соответственно по формулам прямоугольников, трапеций и Симпсона. Следовательно, для любой непрерывной на $[a; b]$ функции $f(x)$ приближенные значения интеграла, полученные по всем этим формулам, стремятся к точному значению при $h \rightarrow 0$.

Оценки погрешностей указанных квадратурных формул позволяют сделать вывод: для достаточно гладкой функции $f(x)$ более точной является квадратурная формула Симпсона.

Для недостаточно гладкой функции целесообразнее применять самую простую квадратурную формулу, например прямоугольников, требующую минимального объема вычислений.

Оценка погрешности по (6.58), (6.62), (6.67) остаточных членов квадратурных формул часто оказывается малоэффективной из-за трудностей, связанных с оценкой производных подынтегральной функции $f(x)$. Поэтому на практике часто пользуются при оценке погрешности приемом, предложенным Рунге. Обозначим через I точное значение интеграла $\int_a^b f(x) dx$, через I_h — его приближенное значение, вычисленное по одной из квадратурных формул с шагом h , через $I_{h/2}$ — при-

ближеннное значение интеграла, вычисленное по той же формуле с шагом $h/2$. Остаточный член каждой квадратурной формулы с шагом h и $h/2$ на основании (6.58), (6.62) и (6.67) можно записать соответственно в виде

$$R_h = h^k M, \quad R_{h/2} = \left(\frac{h}{2}\right)^k M,$$

где k — порядок точности формулы; M — произведение постоянной на производную $f^{(k)}(\xi)$.

Вычислим приближенное значение интеграла по одной и той же квадратурной формуле сначала с шагом h , а затем с шагом $h/2$. Получим

$$I = I_h + h^k M, \quad I = I_{h/2} + \left(\frac{h}{2}\right)^k M.$$

Вычтем эти равенства:

$$I_{h/2} - I_h = M \left(\frac{h}{2}\right)^k (2^k - 1).$$

Получим оценку погрешности по методу Рунге

$$|R_{h/2}| = |M| \left(\frac{h}{2}\right)^k = \frac{|I_{h/2} - I_h|}{2^k - 1}. \quad (6.68)$$

Пользуясь формулой (6.68), можно уточнить приближенное значение интеграла, полагая

$$I = I_{h/2} + R_{h/2} = I_{h/2} + \frac{I_{h/2} - I_h}{2^k - 1}. \quad (6.69)$$

Формулу (6.69) называют формулой экстраполяции по Ричардсону.

Учитывая порядок точности квадратурных формул, выпишем приближенную оценку погрешности по методу Рунге для формул прямоугольников и трапеций ($k = 2$):

$$R_{h/2} \approx \frac{I_{h/2} - I_h}{3} \quad (6.70)$$

и по формуле Симпсона ($k = 4$):

$$R_{h/2} \approx \frac{I_{h/2} - I_h}{15}. \quad (6.71)$$

Прием многократного уменьшения шага и оценки погрешности (6.68) можно запрограммировать и получить алгоритм автоматического выбора шага h для приближенного вычисления интеграла с заданной точностью.

Квадратурная формула Гаусса. Рассмотрим задачу об оптимальном выборе узлов x_i и коэффициентов A_i в квадратурной формуле

$$\int_a^b f(x) dx \approx \sum_{i=1}^n A_i f(x_i),$$

чтобы она была точной для всякого многочлена наивысшей возможной степени $2n - 1$. Отрезок интегрирования $[a; b]$ линейным преобразованием приведем к $[-1; 1]$. Формула (6.52) содержит $2n$ неизвестных A_i и x_i ($i = 1, 2, \dots, n$). По узлам x_i построим многочлен $P_n(x) = (x - x_1)(x - x_2)\dots(x - x_n)$. Чтобы формула

$$\int_{-1}^1 f(x) dx = \sum_{i=1}^n A_i f(x_i) \quad (6.72)$$

имела степень точности $2n - 1$, необходимо и достаточно, чтобы она удовлетворяла условиям интерполяции в узлах x_i и многочлен $P_n(x)$ был ортогональным на $[-1; 1]$ с постоянным весом ко всякому многочлену степени, меньшей n [24]. Указанным условиям удовлетворяют полиномы Лежандра (5.62)

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n.$$

Узлы x_i квадратурной формулы (6.72) являются нулями полинома Лежандра $P_n(x)$, т. е. определяются из условий

$$P_n(x_i) = 0 \quad (i = 1, 2, \dots, n). \quad (6.73)$$

Коэффициенты A_i вычисляются по формулам [2]

$$A_i = \frac{\int_{-1}^1 (x - x_1)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_n) dx}{(x_i - x_1)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_n)},$$

или

$$A_i = \frac{2}{(1 - x_i)^2 [P'_n(x_i)]^2} \quad (i = 1, 2, \dots, n). \quad (6.74)$$

Квадратурная формула (6.72), узлы которой x_i являются нулями полинома Лежандра, а коэффициенты A_i вычисляются по формулам (6.74), называется квадратурной формулой Гаусса, имеющей высокую точность при сравнительно малом числе узлов.

Формула Гаусса для отрезка интегрирования $[a; b]$ может быть получена из (6.72) подстановкой

$$t_i = \frac{b-a}{2} + \frac{b-a}{2} x_i \quad (i = 1, 2, \dots, n), \quad (6.75)$$

тогда

$$\int_a^b f(t) dt = \frac{b-a}{2} \sum_{i=1}^n A_i f(t_i). \quad (6.76)$$

Эту формулу целесообразно применять только в случае хорошей гладкости функции $f(x)$ на отрезке $[a; b]$. Неудобства ее применения состоят в том, что нули x_i и коэффициенты A_i , вообще говоря, — иррациональные числа. Составная квадратурная формула Гаусса в практических вычислениях используется крайне редко.

Квадратурная формула Гаусса является формулой открытого типа. Формула прямоугольников является частным случаем формулы Гаусса при $n = 1$.

Остаточный член квадратурной формулы Гаусса (6.76) с n узлами [4]

$$R_n = \frac{(b-a)^{2n+1} (n!)^4 f^{(2n)}(\xi)}{[(2n!)^3 (2n+1)]}, \quad \xi \in [a; b]. \quad (6.77)$$

Для частных случаев $n = 2, 3, 4$ получим:

$$R_2 = \frac{1}{135} \left(\frac{b-a}{2} \right)^5 f^{IV}(\xi);$$

$$R_3 = \frac{1}{15750} \left(\frac{b-a}{2} \right)^7 f^{VI}(\xi);$$

$$R_4 = \frac{1}{3472875} \left(\frac{b-a}{2} \right)^9 f^{VIII}(\xi).$$

В табл. 6.1 помещены значения узлов x_i и весов A_i формул Гаусса для $n \leq 8$. Значения узлов t_i для любого отрезка $[a; b]$ определяются по формуле (6.75), а затем вычисляются значения подынтегральной функции $f(t_i)$ для квадратурной формулы (6.76).

Численное интегрирование с помощью сплайнов. Рассмотрим квадратурную формулу, использующую кубическую сплайн-интерполяцию подынтегральной функции $f(x)$ на отрезке $[x_i; x_{i+1}]$. Выберем кусочно-полиномиальное представление сплайна (5.102)

$$S_3(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3,$$

где $x \in [x_i; x_{i+1}]$, интерполирующую функцию $f(x)$ в узлах x_i ($i = 0, 1, 2, \dots, N$).

Таблица 6.1

n	x_i	A_i
1	$x_1 = 0$	$A_1 = 2$
2	$-x_1 = x_2 = 0,57735027$	$A_1 = A_2 = 1$
3	$-x_1 = x_3 = 0,77459667$ $x_2 = 0$	$A_1 = A_3 = 0,555555556 = \frac{5}{9}$ $A_2 = 0,88888889 = \frac{8}{9}$
4	$-x_1 = x_4 = 0,86113631$ $-x_2 = x_3 = 0,33998104$	$A_1 = A_4 = 0,34785484$ $A_2 = A_3 = 0,65214516$
5	$-x_1 = x_5 = 0,90617985$ $-x_2 = x_4 = 0,53846931$ $x_3 = 0$	$A_1 = A_5 = 0,23692688$ $A_2 = A_4 = 0,47862868$ $A_3 = 0,56888889$
6	$-x_1 = x_6 = 0,93246951$ $-x_2 = x_5 = 0,66120939$ $-x_3 = x_4 = 0,23861919$	$A_1 = A_6 = 0,17132450$ $A_2 = A_5 = 0,36076158$ $A_3 = A_4 = 0,46791394$
7	$-x_1 = x_7 = 0,94910791$ $-x_2 = x_6 = 0,74153119$ $-x_3 = x_5 = 0,40584515$ $x_4 = 0$	$A_1 = A_7 = 0,12948496$ $A_2 = A_6 = 0,27970540$ $A_3 = A_5 = 0,38183006$ $A_4 = 0,41795918$
8	$-x_1 = x_8 = 0,96028986$ $-x_2 = x_7 = 0,79666648$ $-x_3 = x_6 = 0,52553242$ $-x_4 = x_5 = 0,18343464$	$A_1 = A_8 = 0,10122854$ $A_2 = A_7 = 0,22238104$ $A_3 = A_6 = 0,31370664$ $A_4 = A_5 = 0,36268378$

Запишем приближенную формулу

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx \int_{x_i}^{x_{i+1}} S_3(x) dx = \int_{x_i}^{x_{i+1}} (a_{i0} + a_{i1}(x - x_i) + \\ + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3) dx = a_{i0}h_i + \\ + a_{i2}\frac{h_i^2}{2} + a_{i2}\frac{h_i^3}{3} + a_{i3}\frac{h_i^4}{4}. \quad (6.78)$$

Учитывая выражение (5.108) для коэффициентов сплайна

$$a_{i0} = f(x_i), \quad a_{i1} = \frac{f(x_{i+1}) - f(x_i)}{h_i} - \frac{h_i}{6}(2M_i + M_{i+1}), \\ a_{i2} = \frac{M_i}{2}, \quad a_{i3} = \frac{M_{i+1} - M_i}{6h_i},$$

формулу (6.78) после приведения подобных преобразуем к виду

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx h_i \frac{f(x_i) + f(x_{i+1})}{2} - \frac{h_i^3}{24}(M_i + M_{i+1}), \quad (6.79)$$

где $h_i = x_{i+1} - x_i$, $M_i = S''(x_i)$.

Просуммируем интегралы (6.79) по всем частичным отрезкам, полагая $i = 0, 1, \dots, N-1$, $a = x_0$, $b = x_N$. Получим сплайн-квадратуру

$$\int_a^b f(x) dx \approx \frac{1}{2} \sum_{i=0}^{N-1} h_i (f(x_i) + f(x_{i+1})) - \frac{1}{24} \sum_{i=0}^{N-1} h_i^3 (M_i + M_{i+1}). \quad (6.80)$$

Здесь первое слагаемое в случае постоянного шага есть составная формула трапеций (6.61), второе слагаемое определяет поправку к ней. Если производная $f''(x)$ непрерывна на $[a; b]$, то для поправочного члена справедливо соотношение

$$-h_i^3 \frac{M_i + M_{i+1}}{24} = -h_i^3 \frac{S''(x_i) + S''(x_{i+1})}{24} \approx -\frac{h_i^3}{12} f''(\xi_i),$$

где $\xi_i \in [x_i; x_{i+1}]$.

Таким образом, поправочный член сплайн-квадратуры (6.80) аппроксимирует ошибку (6.60) формулы трапеций. Здесь моменты M_i вычисляются с помощью алгоритмов, описанных в гл. 5.

Если для сплайна $S_3(x)$ воспользоваться его представлением (5.120) через наклоны $m_i = S'_i(x_i)$ ($i = 0, 1, \dots, N$),

то аналогично получаем сплайн-квадратуру в виде

$$\int_a^b f(x) dx \approx \frac{1}{2} \sum_{i=0}^{N-1} h_i (f(x_i) + f(x_{i+1})) + \frac{1}{12} \sum_{i=0}^{N-1} h_i^2 (m_i - m_{i+1}) \quad (6.81)$$

Если сетка равномерна, то эта формула упрощается к виду

$$\int_a^b f(x) dx \approx \frac{h}{2} (f(a) + f(b)) + h \sum_{i=1}^{N-1} f(x_i) + \frac{h^2}{12} (m_0 - m_N). \quad (6.82)$$

Если сплайн $S_3(x)$ удовлетворяет на отрезке $[a; b]$ краевому условию I-го типа, то $m_0 = f'(a)$ и $m_N = f'(b)$; при этом не нужно определять параметры сплайна. При краевых условиях других типов m_0 и m_N можно заменить конечными разностями.

Если $f(x)$ имеет на отрезке $[a; b]$ непрерывную производную $f^{IV}(x)$, то сплайн-квадратуры (6.80) и (6.82) обеспечивают на равномерной сетке точность порядка $O(h^4)$.

6.4. Лабораторная работа 9. Численное интегрирование функции одной переменной

Изложим реализацию вычисления определенных интегралов по описанным в настоящей главе квадратурным формулам на программируемых микрокалькуляторах.

Программа 29 вычисления определенного интеграла по составной квадратурной формуле прямоугольников (6.57) (формуле средних):

C_x	ПД	ИПВ	ИПА	$-$	ИПО	\div	ПС	2	\div
ИПА	$+$	ПА	ПП	23	ИПД	$+$	ПД	ИПС	FL0
10	\times	C/P	\dots	\dots	\dots	\dots	\dots	B/O	
<hr style="border-top: 1px solid black; border-bottom: none; margin: 5px 0;"/> <hr style="border-top: 1px solid black; border-bottom: 1px solid black; margin: 5px 0;"/>									
Подпрограмма вычисления $f(x)$									

Распределение регистров памяти:

Номер регистра	0	A	B	C	D
Переменная	n , счетчик цикла	$a, x_i + \frac{h}{2}$	b	$h = \frac{b-a}{n}$	$\sum f\left(x_i + \frac{h}{2}\right)$

Инструкция. 1. Составить подпрограмму вычисления значений $f(x)$. Значение x хранится в РХ и РА, свободны регистры 1—9.

2. В режиме программирования (F ПРГ) ввести программу.

3. В режиме автоматической работы (F АВТ) ввести исходные данные: (n) ПО (a) ПА (b) ПВ

4. Нажать клавиши В/О С/П, после останова прочесть приближенное значение интеграла.

Контрольный пример. Вычислить значение определенного интеграла $\int_0^1 \sqrt{1+x^4} dx$ по квадратурной формуле прямоугольников (6.57)^o, полагая $n = 6$.

Подпрограмма вычисления $f(x)$:

FX² FA² 1 + FV B/O.

Нажмем клавиши F ПРГ и клавиши программы. Далее введем исходные данные: F АВТ 0 ПА 1 ПВ 6 ПО. Нажав клавиши В/О С/П, через 35 с прочтем ответ:

$$\int_0^1 \sqrt{1+x^4} dx \approx 1,0877925.$$

Для иллюстрации точности вычислений по квадратурной формуле прямоугольников (6.57) вычислим значение интеграла:

$$\int_0^1 \frac{dx}{1+x} = \ln 2 = 0,69314717.$$

Вычисления по программе 29 дают результат $\int_0^1 \frac{dx}{1+x} \approx 0,6922843$,

т. е. два десятичных знака совпадают с точным значением.

Программа 30 вычисления определенного интеграла $\int_a^b f(x) dx$ по составной квадратурной формуле трапеций

ИПВ	ИПА	—	ИПС	÷	ПЗ	ПП	27	0	+
П2	ИПА	ИПЗ	+	ПА	ПП	27	↑	ИП2	+
FLO	09	ИПЗ	×	2	÷	C/P	...		B/O

Подпрограмма $f(x)$

Распределение регистров памяти:

Номер регистра	0	2	3	A	B	C
Переменная	счетчик цикла	$\sum A_i f(x_i)$	$h = \frac{b-a}{n}$	a, x	b	n

Инструкция. 1. Составить подпрограмму вычисления значения $f(x)$. Значение x хранится в РА.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В режиме автоматической работы (F АВТ) ввести исходные данные: (a) ПА (b) ПВ (n) ПС ПО.

4. Нажать клавиши В/О С/П. По окончании вычислений прочесть на индикаторе приближенное значение интеграла.

Контрольный пример. Вычислить $\int_0^1 \sqrt{1+x^4} dx$ по квадратурной формуле трапеций (6.61), полагая $n = 6$. Оценить погрешность по правилу Рунге двойным пересчетом, положив $n = 12$.

Подпрограмма вычисления $f(x)$:

ИПА FX² FX² 1 + FV B/O.

После ввода программы введем исходные данные:

F АВТ 0 ПА 1 ПВ 6 ПС ПО

Нажав клавиши В/О С/П, через 40 с прочтем ответ:

$$\int_0^1 \sqrt{1+x^4} dx \approx 1,0927029.$$

Повторим вычисления при $n = 12$:

0 ПА 1 ПВ 12 ПС ПО, В/О С/П

После вычислений (примерно 75 с) прочтем результат:

$$\int_0^1 \sqrt{1+x^4} dx \approx 1,0902478.$$

По формуле (6.70) определим приближенную оценку погрешности по Рунге: $R_{h/2} = -8,1836666 \cdot 10^{-4}$. Экстраполяцией по Ричардсону уточним значение интеграла $J \approx 1,0894294 \approx 1,089$. Результат округлили до трех десятичных знаков.

Вычисляя значение интеграла $\int_0^1 \frac{dx}{1+x}$ по программе 30 при $n = 6$,

получим результат $I \approx 0,6948773$ (совпадает с точным значением два десятичных знака).

Сравнение аналогичного результата для формулы прямоугольников свидетельствует о двустороннем приближении интеграла с помощью квадратурных формул прямоугольников и трапеций.

Программа 31 вычисления определенного интеграла по составной квадратурной формуле Симпсона (6.66):

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ИПВ	6L	12	×	12	24	÷	13
01	ИПА	6—	13	ИП2	62	25	С/П	50
02	—	11	14	ПП	53	26	+	10
03	ИПС	6C	15	26	26	27	П2	42
04	÷	13	16	↑	0E	28	ИПА	6—
05	П3	43	17	ИП2	62	29	ИПЗ	63
06	ПП	53	18	+	10	30	+	10
07	32	32	19	FL0	5—	31	ПА	4—
08	0	00	20	09	09	32
09	ПП	53	21	ИПЗ	63
10	26	26	22	×	12
11	4	04	23	3	03	...	В/О	52

Распределение регистров памяти:

Номер регистра	0	2	3	A	B	C
Переменная	n , счетчик цикла	$\sum A_i f(x_i)$	$h = \frac{b-a}{n}$	a, x	b	$2n$

Инструкция. 1. Составить подпрограмму вычисления значений $f(x)$. Значение x хранится в РА.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В режиме автоматической работы (F АВТ) ввести исходные данные:

(a) ПА (b) ПВ ($2n$) ПС (n) ПО

4. Нажать клавиши В/О С/П. После ОСТАНОВа пропустить на индикаторе ответ.

Контрольный пример. Вычислить $\int_0^1 \sqrt{1+x^4} dx$ по составной квадратурной формуле Симпсона (6.66) при $n = 6$ и $n = 12$. Оценить погрешность по правилу Рунге.

Подпрограмма вычисления $f(x)$: ИПА Fx² Fx² 1 + F V⁻
 В/О. После ввода программы введем исходные данные: F АВТ 0
 ИА 1 ПВ 6 ПС 3 ПО В/О С/П. Через 40 с получим результат:
 $\int_0^1 \sqrt{1+x^4} dx \approx 1,0894286.$

Повторим вычисления при $n = 12$. После вычислений (при мерно через 70 с) получим: $\int_0^1 \sqrt{1+x^4} dx \approx 1,0894293.$

Приближенная оценка погрешности (6.71) по методу Рунге $R_{h/2} \approx 4,6666666 \cdot 10^{-8}$.

Экстраполяцией по Ричардсону уточним значение интеграла $I \approx 1,089429$. Результат округлен до шести десятичных знаков.

Вычисления по программе 31 интеграла $\int_0^1 \frac{dx}{1+x}$ при $n = 6$ дает результат: $I \approx 0,69316973$ (с точным значением совпадают 4 десятичных знака).

Программа 32 вычисления определенного интеграла по квадратурной формуле Гаусса (6.76) для числа узлов $n \leq 8$:

Адрес ^с	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	СХ	0	13	ИП6	66	26	ИП9	69
01	ПД	4	14	Х	12	27	+	10
02	ИП6	66	15	+	10	28	КИП0	Г0
03	ИП3	63	16	ПП	53	29	Х	12
04	—	11	17	38	38	30	ИПД	6Г
05	2	02	18	П9	49	31	+	10
06	÷	13	19	ИП3	63	32	ПД	4Г
07	П6	46	20	КИП↑	ГЕ	33	FL0	5Г
08	ИП3	63	21	ИП6	66	34	11	11
09	+	10	22	Х	12	35	ИП6	66
10	П3	43	23	—	11	36	Х	12
11	ИП3	63	24	ПП	53	37	С/П	50
12	КИП0	Г0	25	38	38	38	... } f(x)	...
							... B/O	52

Распределение регистров памяти:

Номер регистра	0	1,2; 4,5; 7,8 А, В	3	6	Д
Переменная	счетчик цикла	$-x_i, A_i$	$a, \frac{b+a}{2}$	$b, \frac{b-a}{2}$	$\sum A_i f(x_i)$

Инструкция. 1. Составить подпрограмму вычисления значений $f(x)$. Значение аргумента хранится в регистре X, свободен регистр C.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В режиме автоматической работы (F АВТ) ввести исходные данные: (3k) ПО (a) П3 (b) П6 (A_1) П1 ($-x_1$) П2 (A_2) П4 ($-x_2$) П5 (A_3) П7 ($-x_3$) П8.

Здесь $k = \frac{n}{2}$, если n четно, либо $k = \frac{n+1}{2}$, если n нечетно.

При вводе A_k , $-x_k$ регистры с номерами, кратными трем, пропускаются. Последней вводится пара A_k , x_k , при этом, если k нечетно, то вместо A_k следует вводить $A_k/2$.

4. Осуществить пуск программы В/О С/П.

После окончания вычислений прочесть на индикаторе результат.

Контрольный пример. Вычислить $\int_0^1 \sqrt{1+x^4} dx$ по квадратурной

формуле Гаусса (6.76), полагая $n = 3, 5, 8$.

Программа вычисления $f(x)$: FX² FX² 1 + FV - В/О. После ввода программы при $n = 3$ введем исходные данные: F АВТ 6 ПО 0 П3 1 П6 0,55555555 П1 0,7745967 П2 0,44444444 П4 0 П5.

Нажав клавиши В/О С/П, через 25 с прочтем ответ

$$\int_0^1 \sqrt{1+x^4} dx \approx 1,0894588.$$

При $n = 5$ ввод исходных данных

9	П0	0	П3	1	П6		
0.2369269	П1			0.9061798	П2		
0.4796287	П4			0.5384693	П5		
0.2844444	П7	0	П8			В/О	С/П

Через 35 с прочтем ответ:

$$\int_0^1 \sqrt{1+x^4} dx \approx 1,0894299.$$

При $n = 8$ ввод исходных данных

12	П0	0	П3	1	П6		
0.1012285	П1	0.9602899	П2				
0.2223810	П4	0.7966665	П5				

0.3137066 П7 0.5255324 П8
 0.3626838 ПА 0.1834346 ПВ В/О С/П

Через 50 с получим результат

$$\int_0^1 \sqrt{1+x^4} dx \approx 1,0894293.$$

По квадратурной формуле Гаусса получим при $n = 6$ значение определенного интеграла $\int_0^1 \frac{dx}{1+x} \approx 0,69314715$, т. е. семь десятичных знаков совпадают с точным значением интеграла.

Анализ полученных результатов свидетельствует о том, что для функции $f(x)$, имеющей производные высоких порядков, формула Гаусса дает результаты намного точнее, чем остальные квадратурные формулы. Для менее гладких функций целесообразнее использовать квадратурные формулы прямоугольников или трапеций.

Выбор шага интегрирования для достижения заданной точности вычислений можно проводить для каждой квадратурной формулы по оценке ее остаточного члена. В рассмотренном выше примере

$$f(x) = \sqrt{1+x^4}; \quad f''(x) = \frac{6x^2 - 2x^6}{(1+x^4)^{3/2}}.$$

Максимальное значение второй производной на отрезке $(0; 1)$ равно 2, т. е. $f''(x) \leq 2$ при $x \in [0; 1]$. Для квадратурной формулы прямоугольников получим оценку остаточного члена

$$|R| = \frac{(b-a)h^2}{24} |f''(\xi)| \leq \frac{h^2 2}{24} < 0,08h^2.$$

При числе узлов $n = 6$ имеем $h \approx 0,17$, поэтому $|R| < 0,0023 < 0,01$, т. е. два десятичных знака будут точными. Для достижения точности, например, $\epsilon = 10^{-4}$, учитывая погрешность округлений, выберем величину шага из условия $|R| \leq 0,5 \cdot 10^{-4}$, отсюда $0,08h^2 \leq 0,5 \cdot 10^{-4}$.

Таким образом, для достижения указанной точности нужно выбрать шаг $h < 0,025$, т. е. $n = \frac{b-a}{h} = 40$, вычисления проводить с пятью знаками после запятой. Аналогично оценивается погрешность и осуществляется выбор шага для квадратурных формул трапеций и Симпсона.

Задания к лабораторной работе 9.

Поставить переключатель Р—Г в положение Р.

Вычислить приближение определенные интегралы:

а) по формуле Симпсона, полагая $2n = 6$;

б) по формуле Симпсона, полагая $2n = 12$, оценить погрешность по правилу Рунге, уточнить результат;

в) по формуле прямоугольников или трапеций с точностью $\epsilon = 10^{-2}$, определяя величину шага h по оценке остаточного члена;

г) по квадратурной формуле Гаусса, полагая $n = 3$;

д) сравнить результаты, полученные по различным квадратурным формулам, и время вычислений:

$$1) \int_0^1 \sqrt{1+x^3} dx;$$

$$2) \int_1^3 \sqrt{1+x^3} dx;$$

$$3) \int_1^2 \frac{e^x}{x} dx;$$

$$4) \int_1^2 \frac{\sin x}{x} dx;$$

$$5) \int_1^2 \frac{\cos x}{x} dx;$$

$$6) \int_2^3 \frac{dx}{\ln x};$$

$$7) \int_0^1 \sqrt{1+\sin^2 x} dx;$$

$$8) \int_1^3 \sqrt{1+x^4} dx;$$

$$9) \int_0^1 \sin x^2 dx;$$

$$10) \int_0^1 \cos x^2 dx;$$

$$11) \int_0^1 e^{-x^2} dx;$$

$$12) \int_0^{\frac{\pi}{2}} \sqrt{2+\cos x} dx;$$

$$13) \int_0^{\frac{\pi}{2}} \sqrt{2+\sin x} dx;$$

$$14) \int_0^1 \cos e^x dx;$$

$$15) \int_0^1 \sin e^x dx;$$

$$16) \int_1^2 \sqrt{1+\ln x} dx;$$

$$17) \int_1^e \sqrt{1+\ln x} dx;$$

$$18) \int_0^{\frac{\pi}{2}} e^{\sin x} dx;$$

$$19) \int_0^{\frac{\pi}{2}} e^{\cos x} dx;$$

$$20) \int_1^2 \sqrt{\ln x} dx;$$

$$21) \int_0^1 \sqrt{x} e^x dx;$$

$$22) \int_0^1 \sqrt{x} \cos x dx;$$

$$23) \int_0^1 \sqrt{x} \sin x dx;$$

$$24) \int_2^3 \frac{\sqrt{x}}{\cos x} dx;$$

$$25) \int_0^1 \sqrt{x+x^3} dx;$$

$$26) \int_0^1 \sqrt{1+x^6} dx;$$

$$27) \int_0^1 e^{-x^2} dx;$$

$$28) \int_{-1}^0 e^{-x^2} dx;$$

$$29) \int_2^3 \frac{x}{\ln x} dx;$$

$$30) \int_1^2 e^{\frac{1}{x}} dx.$$

Глава 7. ЧИСЛЕННЫЕ МЕТОДЫ МИНИМИЗАЦИИ ФУНКЦИЙ

7.1. Постановка задачи

Многие инженерные и математические задачи сводятся к нахождению минимума функции нескольких переменных. Основной целью решения задач управлении некоторыми отрас-

лями промышленности, сельского хозяйства, транспорта обычно является достижение некоторого оптимального режима работы. В этих случаях задачи минимизации обычно называют задачами оптимизации, а минимизируемую функцию — целевой функцией.

Пусть функция $f(x) = f(x_1, x_2, \dots, x_n)$ n переменных определена и непрерывна в области $D \subset R^n$. Точка $x^* = (x_1^*, x_2^*, \dots, x_n^*) \in D$ называется точкой локального минимума (максимума) функции $f(x)$, если существует некоторая ε — окрестность точки x^* , такая, что для всех $x = (x_1, x_2, \dots, x_n)$ из этой окрестности выполняются неравенства

$$\begin{aligned} f(x) &\geq f(x^*) \quad (f(x) \leq f(x^*)), \\ |x_i - x_i^*| &< \varepsilon, \quad i = 1, 2, \dots, n. \end{aligned} \quad (7.1)$$

Необходимые условия экстремума дифференцируемой функции $f(x)$, как известно, можно представить в виде системы n нелинейных уравнений

$$\frac{\partial f}{\partial x_1} = 0, \quad \frac{\partial f}{\partial x_2} = 0, \quad \dots, \quad \frac{\partial f}{\partial x_n} = 0. \quad (7.2)$$

Точки, определяющие решение системы (7.2), называются стационарными.

Если в окрестности $\rho(x^0)$ стационарной точки $x^0 = (x_1^0, \dots, x_n^0)$ функция $f(x)$ является выпуклой, т. е. для любого $0 < \alpha \leq 1$

$$f(\alpha x^1 + (1 - \alpha)x^2) \leq \alpha f(x^1) + (1 - \alpha)f(x^2), \quad x^1, x^2 \in \rho(x^0), \quad (7.3)$$

то стационарная точка x^0 является точкой минимума. В частности, если в окрестности $\rho(x^0)$ стационарной точки x^0 функция $f(x)$ дважды непрерывно дифференцируема и ее матрица Гессе

$$I(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \quad (7.4)$$

положительно определена в x^0 , то функция $f(x)$ будет выпуклой, а x^0 — точкой минимума.

Чтобы симметрическая матрица $A = [a_{ij}]$ ($a_{ii} = a_{ji}$) была положительно определенной, необходимо и достаточно, чтобы каждый из ее главных миноров $a_{11}, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \dots, \det A$ был положителен (критерий Сильвестра).

Решение системы нелинейных уравнений (7.2) и проверка положительной определенности матрицы (7.4) требуют большого объема вычислений. При этом часто встречаются недифференцируемые функции, либо функции, производные которых записываются громоздкими аналитическими выражениями. Поэтому более эффективными являются численные методы решения задачи, не требующие проверки условий (7.3), (7.4).

Численное решение задач минимизации сводится к построению итераций

$$x^{n+1} = x^n - d^n$$

таких, что последовательность $\{x^1, \dots, x^n, \dots\}$ является минимизирующей, т. е.

$$\lim_{n \rightarrow \infty} x^n = x^*, \quad \min_{x \in D} f(x) = f(x^*).$$

В зависимости от характера построения d^n численные методы разделяются на прямые (без использования производных), методы спуска (с использованием первых производных) и методы с использованием вторых производных. Методы минимизации изложены в литературе [35, 6].

7.2. Одномерная минимизация функции

Рассмотрим численные методы нахождения минимума функции $f(x)$ одной действительной переменной. Предположим, что на отрезке $[a; b]$ функция $f(x)$ унимодальна, т. е. существует единственная точка $x^* \in [a; b]$, в которой функция принимает минимальное значение, и $f(x)$ строго убывает для $x < x^*$ и строго возрастает для $x > x^*$. Унимодальная функция $f(x)$ может быть недифференцируемой на $[a; b]$. Важно, чтобы она была только определена в каждой точке отрезка. Стратегия поиска минимума заключается в выборе такой последовательности точек x_1, x_2, \dots, x_n , что при некотором i точка минимума x^* принадлежит отрезку $[x_{i-1}; x_{i+1}]$. Интервал $[x_{i-1}; x_{i+1}]$ называют интервалом неопределенности. Поиск продолжается до тех пор, пока длина интервала неопределенности не станет меньше заданного числа ϵ .

Для вычисления минимума функции $f(x)$ на микрокалькуляторе целесообразно использовать методы, не требующие вычисления производных, в частности, методы деления отрезка пополам и золотого сечения. Они сходятся медленно, но просты в реализации.

Метод деления отрезка пополам. Пусть $f(x)$ унимодальная на $[a; b]$ и $x^* \in (a; b)$ — точка минимума $f(x)$. Выберем достаточно малое положительное число δ , $0 < \delta < b - a$ и определим точки

$$x_1 = \frac{a + b - \delta}{2}, \quad x_2 = \frac{a + b + \delta}{2}. \quad (7.5)$$

Вычислим значения $f(x_1)$, $f(x_2)$ и сравним их между собой. Если $f(x_1) < f(x_2)$, то полагаем $a_1 = a$, $b_1 = x_2$, иначе полагаем $a_1 = x_1$, $b_1 = b$. Из определения унимодальной функции следует, что $x^* \in [a_1; b_1]$. Заменим в формулах (7.5) a и b на a_1 и b_1 соответственно, вычислим новые точки x_3 , x_4 и новый отрезок $[a_2; b_2]$ такой, что $x^* \in [a_2; b_2]$. После повторения таких вычислений n раз найдем отрезок $[a_n; b_n]$ длины

$$b_n - a_n = \frac{b - a}{2^n} + \left(1 - \frac{1}{2^n}\right)\delta,$$

содержащий точку минимума x^* . Если в качестве точки минимума принять середину этого отрезка, т. е. положить

$$x^* \approx x_n^* = \frac{a_n + b_n}{2}, \quad (7.6)$$

то абсолютная погрешность значения x^*

$$|x^* - x_n^*| \leq \frac{b_n - a_n}{2} < \frac{b - a}{2^{n+1}} + \delta. \quad (7.7)$$

Для получения значения x^* с точностью ε количество n делений отрезка при любом $0 < \delta < \varepsilon$ выбираем из условия

$$\frac{b - a}{2^{n+1}} + \delta < \varepsilon. \quad (7.8)$$

Метод деления отрезка пополам всегда приводит к определению точки локального минимума с заданной точностью, но он требует большого числа вычислений функции $f(x)$.

Метод золотого сечения. Метод применим к унимодальным на отрезке $[a; b]$ функциям. Золотым сечением отрезка $[a; b]$ называется деление его точкой c на две неравные части так, чтобы отношение всего отрезка к большей части равнялось отношению большей части к меньшей, т. е. $(b - a) : (b - c) = (b - c) : (c - a)$. Число $\frac{1 + \sqrt{5}}{2} = 1,6180\dots$ называется золотым отношением.

Определим две точки c и d :

$$c = a + \frac{3 - \sqrt{5}}{2} (b - a) \approx 0,618a + 0,382b,$$
$$d = a + b - c = a + \frac{\sqrt{5} - 1}{2} (b - a) \approx 0,382a + 0,618b,$$
$$c < d.$$

(7.9)

Они реализуют золотое сечение отрезка $[a; b]$, поскольку

$$\frac{b - a}{b - c} = \frac{b - c}{c - a} = \frac{b - a}{d - a} = \frac{d - a}{b - d} = \frac{1 + \sqrt{5}}{2} \approx 1,6180339 \dots$$

Кроме того, точка c осуществляет золотое сечение отрезка $[a; d]$, а точка d — золотое сечение отрезка $[c; b]$.

Если $f(c) \geq f(d)$, то полагаем

$$a_1 = c, \quad b_1 = b, \quad c_1 = d,$$
$$d_1 = a_1 + b_1 - d = 0,382a_1 + 0,618b_1.$$

Если $f(c) < f(d)$, то полагаем

$$a_1 = a, \quad b_1 = d, \quad d_1 = c, \quad c_1 = a_1 + b_1 - c = 0,618a_1 + 0,382b_1.$$

Точка минимума $x^* \in [a_1; b_1]$. Точки c_1 и d_1 определяют золотое сечение отрезка $[a_1, b_1]$. Сравнивая значения функции в точках c_1 и d_1 , определяем отрезок $[a_2; b_2]$ такой, что $x^* \in [a_2; b_2]$ и т. д. При этом $b_1 - a_1 = \frac{\sqrt{5} - 1}{2} (b - a), \dots$, $b_n - a_n = \left(\frac{\sqrt{5} - 1}{2}\right)^n (b - a)$.

Процесс продолжаем до тех пор, пока не получим интервал неопределенности заданной длины 2ϵ . На каждом шаге метода золотого сечения требуется вычислить значение функции только в одной точке, что обеспечивает его преимущество по сравнению с методом деления отрезка пополам.

7.3. Лабораторная работа 10. Одномерная минимизация функции

Приведем описание алгоритма и программу поиска минимума функции $f(x)$ методом золотого сечения. Пусть ϵ — требуемая полуширина интервала неопределенности.

1. Вычислить: $\lambda = \frac{\sqrt{5} - 1}{2}$; $h = \frac{b - a}{2}$; $z = \frac{a + b}{2}$.
2. Если $h < \epsilon$, перейти к п. 6.

Если $h > \epsilon$, вычислить: $g = h\lambda^3$, $u = f(z - g)$, $v = f(z + g)$.

3. Вычислить $\delta = u - v$.

4. Если $\delta = 0$, положить $h = g$; перейти к п. 2; если $\delta \neq 0$, положить $h = h\lambda$.

5. Если $h < \epsilon$, перейти к п. 6. Если $h > \epsilon$, то: вычислить $g = h\lambda^3$; если $\delta < 0$, положить $v = u$, вычислить $u = f(z - g)$; если $\delta > 0$, положить $u = v$, вычислить $v = f(z + g)$; перейти к п. 3.

6. Положить $x^* = z$.

Распределение регистров памяти:

Номер регистра	7	8	9	А	В	С	Д	0—6
Переменная	$(2\epsilon)^2$	λ	$\Delta_n = b_n -$ $- a_n$	a_n	b_n	$f(c_n)$	$f(d_n)$	свободны

Инструкция. Аргумент x для вычисления значения $f(x)$ хранится в регистре X.

(a) ПА (b) ПВ (2 ϵ) В/О С/П.

Программа 33 поиска минимума функции одной переменной методом золотого сечения.

Fx ²	П7	5	FV	1	—	2	÷	П8	ИПВ
ИПА	—	П9	ПП	42	ПП	60	ПП	68	ИП9
/—/	П9	ИПС	ИПД	—	FX ≠ 0	09	FX < 0	36	ИПВ
ИП9	/—/	ПП	44	БП	17	ПП	42	ПП	60
БП	19	ИПА	ИП9	ИП9	FX ²	ИП7	—	Fx < 0	55
F,	2	÷	+	С/П	ИП9	ИП8	×	П9	В/О
ИПА	ИП9	+	ПА	ПП	76	ПС	В/О	ИПВ	ИП9
—	ПВ	ПП	76	ПД	В/О	Подпрограмма $f(x)$			

Контрольный пример. Найти минимум функции $f(x) = \frac{x^2}{2} - \sin x$ на промежутке $[0,5; 1]$ с точностью $\epsilon = 10^{-2}$.

Подпрограмма вычисления $f(x)$:

F sin FBX FX² 2 ÷ — /—/ В/О

После ввода программы и подпрограммы перейдем в режим автоматической работы, введем исходные данные и приступим к вычислениям:

0,5 ПА 1 ПВ 0,02 В/О С/П.

Через 2 мин прочтем результат:

$x^* = 0,738 \approx 0,74$.

ПП Для вычисления значения $f(x^*)$ нажмем клавиши БП 70 ... ПП и прочтем: $f(x^*) = -0,4005$.
7 раз

$$f_{\min} = f(0,74) = -0,4005.$$

Отметим, что при необходимости программу можно сократить, отбросив команды 02—08. При этом значение $\lambda = \frac{\sqrt{5}-1}{2}$ следует вычислить и занести в РБ в автоматическом режиме, все адреса подпрограмм, условных и безусловных переходов уменьшатся на 7.

Задания к лабораторной работе 10. Найти минимум функции $f(x)$ на промежутке $[a; b]$ с точностью $\epsilon = 10^{-2}$.

- | | |
|--|---|
| 1) $f(x) = e^x + x^2$, $[-1; 0]$; | 18) $f(x) = e^{\frac{1}{x}} + \ln x$, $[1; 3]$; |
| 2) $f(x) = e^x + \frac{1}{x}$, $[0,5; 1]$; | 19) $f(x) = e^{-x} + \frac{1}{1-x}$, $[-0,5; 0,5]$; |
| 3) $f(x) = e^x - \ln x$, $[0,3; 1]$; | 20) $f(x) = -\operatorname{tg} x - \frac{1}{x}$,
$x \in [-1; -0,5]$; |
| 4) $f(x) = e^x + \frac{1}{x+1}$, $[-0,5; 0,5]$; | 21) $f(x) = e^x - \operatorname{tg} x - x$, $[0; 1]$; |
| 5) $f(x) = \operatorname{tg} x + \frac{1}{x}$, $[0,5; 1]$; | 22) $f(x) = x^2 - \sin x$, $[0; 1]$; |
| 6) $f(x) = \operatorname{tg} x + e^{-x} + x$, $[-1; 0]$; | 23) $f(x) = e^{-x} + \sin x$, $[-1; 0]$; |
| 7) $f(x) = x^2 + \sin x$, $[-1; 0]$; | 24) $f(x) = x^4 + 2x^2 - 4x$, $[0; 1]$; |
| 8) $f(x) = e^x - \sin x$, $[0; 1]$; | 25) $f(x) = x^2 - xe^{-x}$, $[0; 1]$; |
| 9) $f(x) = x^4 + 2x^2 + 4x$, $[-1; 0]$; | 26) $f(x) = x + \frac{2-x}{x^2}$, $[0,5; 1]$; |
| 10) $f(x) = xe^x + x^2$, $[-1; 0]$; | 27) $f(x) = \frac{1}{x} - \frac{1}{\ln x}$, $[0,3; 0,7]$; |
| 11) $f(x) = \frac{x+2}{x^2} - x$, $[-1; -0,5]$; | 28) $f(x) = \frac{1}{x} + \ln^2 x$, $[1; 3]$; |
| 12) $f(x) = x + \frac{1}{\ln x}$, $[1,5; 2,5]$; | 29) $f(x) = e^{\frac{1}{x}} + \ln x$, $[1; 3]$; |
| 13) $f(x) = x + \ln^2 x$, $[0,3; 1]$; | 30) $f(x) = e^{x-1} + \frac{1}{x}$, $[0,5; 1,5]$. |
| 14) $f(x) = x - \ln \ln x$, $[1,3; 2,3]$; | |
| 15) $f(x) = x + \frac{1}{\operatorname{arctg} x}$, $[0,5; 1,5]$; | |
| 16) $f(x) = e^{-x} + x^2$, $[0; 1]$; | |
| 17) $f(x) = e^{-x} - \frac{1}{x}$, $[-1; -0,5]$; | |

7.4. Метод покоординатного спуска

Наиболее простым из прямых методов поиска минимума функции нескольких переменных является метод покоординатного спуска. Изложим вначале идею метода для случая функции двух переменных $f(x, y)$.

Выберем начальное приближение $M_0(x^0, y^0)$. Зафиксируем y^0 и найдем минимум функции одной переменной $f(x, y^0)$.

Пусть он достигается при $x = x^1$. Вдоль прямой, параллельной оси ОХ, осуществляем спуск в точку $M_1(x^1, y^0)$. Фиксируем x^1 и находим минимум функции одной переменной $f(x^1, y)$. Пусть это будет y^1 . Из точки $M_1(x^1, y^0)$ движемся вдоль прямой, параллельной оси ОY, к точке $M_2(x^1, y^1)$. Затем опять осуществляем спуск из точки M_2 вдоль прямой параллельной оси ОХ и т. д. (рис. 21).

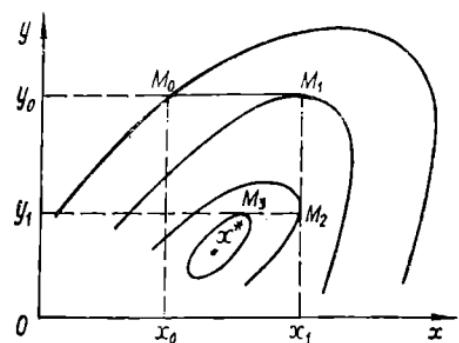


Рис. 21

линией уровня, проходящей через точку $M_0(x^0, y^0)$, выполняются условия [19]:

$$\frac{\partial^2 f}{\partial x^2} \geq a > 0; \quad \frac{\partial^2 f}{\partial y^2} \geq b > 0; \quad \left| \frac{\partial^2 f}{\partial x \partial y} \right| \leq c; \quad ab > c^2. \quad (7.10)$$

Частные производные функции стремятся к нулю:

$$|f'_x(M_k)| \leq q^k |f'_x(M_0)| \rightarrow 0, \quad |f'_y(M_k)| \leq q^k |f'_y(M_0)| \rightarrow 0 \text{ при } k \rightarrow \infty, \quad (7.11)$$

где $0 < q = \frac{c^2}{ab} < 1$, т. е. метод сходится со скоростью геометрической прогрессии.

Известны некоторые модификации метода покоординатного спуска, в частности метод покоординатного спуска с постоянным шагом. В этом случае не требуется на каждом шаге находить минимум функции одной переменной $f(x, y^{k-1})$ или $f(x^k, y)$, а спуск в направлении координатных осей осуществляется в сторону убывания функции с постоянным шагом h , т. е.

$$\begin{aligned} x^{k+1} &= x^k + hi, \quad i = 1, 2, \dots, m; \\ y^{k+1} &= y^k + hr, \quad r = 1, 2, \dots, p. \end{aligned} \quad \} \quad (7.12)$$

Количество шагов спуска m и p по каждой координате выбирается из условия

$$f(x^k + hm, y^k) < f(x^k, y^k); \quad f(x^{k+1}, y^k + hp) < f(x^{k+1}, y^k). \quad (7.13)$$

Как только функция $f(x, y)$ перестает убывать, спуск прекращают.

Метод покоординатного спуска легко обобщается на случай функции n переменных $f(x_1, x_2, \dots, x_n)$. На k -м шаге спуск по координате x_i осуществляется из условия минимума функции одной переменной x_i :

$$f(x_1^k, x_2^k, \dots, x_{i-1}^k, x_i, x_{i+1}^{k-1}, \dots, x_n^{k-1}) \quad (7.14)$$

и найденное значение x_i обозначается x_i^k .

Заметим, что метод покоординатного спуска для минимизации функции n переменных $f(\mathbf{x}) = (\mathbf{A}\mathbf{x}, \mathbf{x}) - 2(\mathbf{b}, \mathbf{x})$, $\mathbf{x} = (x_1, \dots, x_n)$ совпадает с итерационным методом Зейделя для системы линейных уравнений $\mathbf{A}\mathbf{x} = \mathbf{b}$, в то время как метод градиентного спуска (см. п. 7.5) эквивалентен методу простой итерации.

7.5. Метод градиентного спуска

Метод определения минимума функции $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, называемый методом градиентного или наискорейшего спуска, предложен Коши.

Для минимизации по методу спуска выбирается начальная точка $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$ (обычно в соответствии с физическим смыслом задачи). Функция $f(\mathbf{x}) = f(\mathbf{x}^0)$ определяет в n -мерном пространстве гиперповерхность, градиент которой указывает направление наибыстreichшего возрастания функции.

Поэтому в направлении $-\text{grad } f(\mathbf{x}^0)$ функция быстрее всего убывает при бесконечно малом движении из данной точки. Спуск по этому направлению до минимума определяет новое приближение \mathbf{x}^1 . В этой точке снова определяется градиент и осуществляется спуск в направлении антиградиента. Случай $n = 2$ представлен на рис. 22.

Искомый вектор \mathbf{x} последовательно уточняется на k -й итерации метода градиентного спуска по формуле

$$\mathbf{x}^{k+1} = \mathbf{x}^k - h_k \text{grad } f(\mathbf{x}^k), \quad (k = 0, 1, 2 \dots), \quad (7.15)$$

где $\mathbf{x}^k = (x_1^k, x_2^k, \dots, x_n^k)$,

$$\text{grad } f(\mathbf{x}^k) = \left(\frac{\partial f(x_1^k, \dots, x_n^k)}{\partial x_1}, \dots, \frac{\partial f(x_1^k, \dots, x_n^k)}{\partial x_n} \right);$$

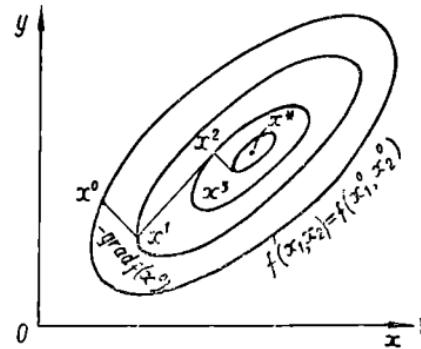


Рис. 22

h_k — оптимальный шаг для k -й итерации, определяемый из условия

$$\varphi(h_k) = \min_{h>0} \varphi(h) = \min_{h>0} f(\mathbf{x}^k - h \operatorname{grad} f(\mathbf{x}^k)). \quad (7.16)$$

Таким образом, на каждом шаге градиентного спуска требуется решать еще задачу минимизации функции одной переменной (7.16) каким-либо численным методом. В частности, можно разложить функцию $\varphi(h)$ в ряд Тейлора, ограничившись членами второго порядка, и определить h_k из условия $\frac{d\varphi}{dh} = 0$. Однако такой метод приводит к очень громоздким вычислениям. При этом необходимо учитывать также трудоемкость вычисления значений функции $f(\mathbf{x})$ и ее градиента в точках x^k . Поэтому на практике часто h_k выбирают эмпирическим путем. Осуществляется спуск при произвольном h_k ; если значение функции $f(\mathbf{x}^{k+1})$ уменьшится, то переходим к следующему шагу спуска, если же $f(\mathbf{x}^{k+1})$ не убывает, то уменьшаем шаг h_k . Следует учитывать, что если h_k выбрать очень малым, то это приводит к существенному увеличению объема вычислений, если h_k слишком большое, то это может привести к проскачиванию через минимум функции. Вычисления по формуле (7.15) проводим до тех пор, пока функция $f(\mathbf{x})$ практически перестанет убывать, т. е. до выполнения для наперед заданного ϵ неравенства

$$|f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)| < \epsilon.$$

Критерием окончания итерационного процесса поиска минимума можно выбрать также условие

$$\max_i \left| \frac{\partial f(\mathbf{x}^k)}{\partial x_i} \right| < \epsilon.$$

Если функция $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$ выпукла в области D , содержащей точку минимума x^* и начальное приближение $\mathbf{x}^0 \in D$, то метод наискорейшего спуска сходится к точке минимума линейно [6], т. е.

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq g^k \|\mathbf{x}^1 - \mathbf{x}^0\|,$$

где $0 < g < 1$.

7.6. Метод Ньютона

Описанным выше методам свойственен один общий недостаток — медленная сходимость, если поверхности (линии) уровня минимизируемой функции вытянуты, сильно отли-

чаются от сфер (окружностей). В методе Ньютона минимизация функции нескольких переменных этот недостаток устраняется учетом значений вторых производных, однако применим этот метод для более узкого класса функций.

Пусть в окрестности стационарной точки \mathbf{x}^* функция $f(\mathbf{x}) = f(x_1, \dots, x_n)$ дважды непрерывно дифференцируема и ее матрица Гессе

$$H(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{pmatrix}$$

положительно определена. Тогда, применяя для решения системы

$$\begin{cases} \frac{\partial f(\mathbf{x})}{\partial x_1} = 0; \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} = 0 \end{cases} \quad (7.17)$$

метод Ньютона (4.37), либо модифицированный метод Ньютона (4.38), получим итерационный процесс для минимизации функции

$$\mathbf{x}^{k+1} = \mathbf{x}^k - H^{-1}(\mathbf{x}^k) \operatorname{grad} f(\mathbf{x}^k), \quad (7.18)$$

либо

$$\mathbf{x}^{k+1} = \mathbf{x}^k - H^{-1}(\mathbf{x}^0) \operatorname{grad} f(\mathbf{x}^k). \quad (7.19)$$

Положительная определенность матрицы $H(\mathbf{x})$ обеспечивает сходимость метода Ньютона к решению системы (7.17), причем сходимость будет квадратичной, а при применении модифицированного метода Ньютона — линейной.

Итерационный процесс (7.18), либо (7.19) называют методом Ньютона (модифицированным методом Ньютона) минимизации функции $f(\mathbf{x})$ n переменных $\mathbf{x} = (x_1, x_2, \dots, x_n)$.

Следует учесть, что в методе Ньютона на погрешность накладывается погрешность обращения матрицы $H(\mathbf{x}^k)$. В связи с этим для функции n переменных при большом n применяют модифицированный метод Ньютона (7.19). Отметим, что для квадратичной формы метод Ньютона дает точный результат при первой итерации.

Вычислительную схему минимизации методом Ньютона рассмотрим на конкретных примерах.

Пример 1. Найти минимум функции

$$f(x, y) = e^x + x^2 + 2xy + 2y^2. \quad (7.20)$$

В качестве начальной рассмотрим точку $O(0; 0)$.

Вычислим $\text{grad } f(x, y)$ и $H(x, y)$:

$$\text{grad } f(x, y) = (e^x + 2x + 2y, 2x + 4y);$$

$$H(x, y) = \begin{pmatrix} e^x + 2 & 2 \\ 2 & 4 \end{pmatrix}.$$

Далее:

$$H^{-1}(x, y) = \frac{1}{4(e^x + 1)} \begin{pmatrix} 4 & -2 \\ -2 & e^x + 2 \end{pmatrix};$$

$$\begin{pmatrix} x \\ y \end{pmatrix} - H^{-1}(x, y) \text{grad } f(x, y) = \begin{pmatrix} \frac{e^x(x-1)}{e^x+1} \\ \frac{e^x(1-x)}{2(e^x+1)} \end{pmatrix}.$$

Формулы (7.18) примут вид:

$$\begin{aligned} x^{k+1} &= \frac{e^{x^k}(x^k - 1)}{e^{x^k} + 1}; \\ y^{k+1} &= -\frac{x^{k+1}}{2}. \end{aligned} \quad (7.21)$$

Программа 34 минимизации функции (7.20) методом Ньютона (7.18)

ИП1	Fe ^X	ПЗ	ИП3	ИП1	1	-	×	ИП3	1
+	÷	П1	2	÷	/-	↑	ИП1	+	
×	2	×	ИП1	FX ²	+	ИП1	Fe ^X	ПЗ	
C/P	БП	03						+	

После ввода в режиме программирования программы введем начальное значение x^0 (y^0 в формулах (7.21) не используется): F АВТ 0 П1 и приступим к вычислениям (время вычислений на каждом шаге около 10 с):

Клавиша	Индикация $f(x, y)$
B/O	0.73153064
C/P	0.72796957
C/P	0.72796903
C/P	0.72796905

Вызовем из регистров 1 и 2 значения x и y :

ИП1 ($x = -0.56714324$) ИП2 ($y = 0.28357162$).

Ответ: $f_{\min} = 0.727969$ при $x = -0.567143$, $y = 0.283572$.

Пример 2. Рассмотрим функцию «овражного» типа

$$f(x, y) = 100(x^2 - y)^2 + (1 - x)^2 \quad (7.22)$$

(тестовую функцию Розенброка [35]). В качестве начальной выберем точку $M_0(-1, 2; 1)$. Область положительной определенности матрицы $H(x, y)$: $y < x^2 + 0,005$. Минимум функция достигает в точке $M^*(1; 1)$.

Отметим, что точка M_0 не попадает в выпуклую окрестность точки M^* , отвечающую условиям применимости метода Ньютона, однако после двух итераций она попадает в требуемую область.

Формулы (7.18) для функции Розенброка:

$$\begin{aligned}x^{k+1} &= \frac{1 - x^k}{200((x^k)^2 - y^k) + 1} + x^k; \\y^{k+1} &= x^k \left(\frac{1 - x^k}{200((x^k)^2 - y^k) + 1} + x^{k+1} \right).\end{aligned}$$

Программа 35 минимизации функции (7.22) методом Ньютона (7.18):

1	ИП1	-	ИП1	FX ²	ИП2	-	2	0	0
×	1	+	÷	ИП1	XУ	↑	ИП1	+	П1
C/P	+	×	P2	C/P	БП	00			

После ввода в режиме программирования программы вводим координаты начальной точки:

F АВТ 1.2 /-/ П1 1 П2

и приступаем к вычислениям (вычисление значения функции программой не предусмотрено):

Клавиша	Индикация x^{k+1}	Клавиша	Индикация
B/O C/P	-1.1752809	C/P	1.3806741
C/P	0.7631266	C/P	-3.1750614
C/P	0.76344139	C/P	0.58284266
C/P	0.99999525	C/P	0.94403376
C/P	0.999995564	C/P	0.99999328
C/P	1	C/P	1
C/P	1	C/P	1

Ответ: функция достигает минимума в точке $M^*(1; 1)$; $f_{\min} = 0$.

7.7. Лабораторная работа 11. Методы минимизации функции нескольких переменных

Программа 36 вычисления минимума функции n переменных ($n \leq 9$) $f(x_1, \dots, x_n)$ методом покоординатного спуска с постоянным шагом h по формулам (7.12):

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	(n)	0n	14	ИПС	6-	28	ИПА	6-
01	ПО	40	15	ПП	53	29	-	11
02	ПП	53	16	19	19	30	$Fx > 0$	59
03	38	38	17	БП	51	31	14	14
04	ПП	53	18	08	08	32	ИПД	6-
05	19	19	19	ПА	4-	33	/-/	0L
06	ПП	53	20	КИП↑	E	34	ПД	4-
07	22	22	21	ПВ	4L	35	ИПВ	6L
08	ИПА	6-	22	ИПД	6-	36	KP↑	LE
09	С/П	50	23	+	10	37	B/O	52
10	FLO	5-	24	KP↑	L E	38	... } f(x)	...
11	04	04	25	ПП	53
12	БП	51	26	38	38		B/O	52
13	00	00	27	ПС	4-			

Распределение регистров памяти :

Номер регистра	0	1...9	A	B	C	D
Переменная	счетчик цикла	$x_1 \dots x_n$	$f(x)$	x'_i	$f(x')$	h

Инструкция. 1. Составить подпрограмму вычисления $f(x)$. Значения x_1, \dots, x_n хранятся в регистрах $1 \div n$ ($n \leq 9$).

2. В режиме программирования (F ПРГ) ввести программу.

3. Ввести начальные данные: F АВТ $(x_1^{(0)})$ П1 ... $(x_n^{(0)})$ П(n) (h) ПД.

4. Приступить к вычислениям, нажав клавиши В/О С/П. После останова индицируется минимум f по переменной x_n .

Для продолжения вычислений нажимаем клавишу С/П; после каждого ОСТАНОВа индицируется минимум функции f соответственно по переменным: $x_{n-1}, x_{n-2}, \dots, x_1, x_n, x_{n-1}, \dots$.

Если при заданном шаге h значение функции f перестает уменьшаться, то можно продолжить поиск минимума при меньшем шаге h_1 , нажав клавиши (h_1) ПД ИПА С/П. Если после останова содержимое регистра x изменилось (в результате, например, вызова из памяти значения x_i либо при вводе нового значения h), то для продолжения вычислений следует нажать клавиши ИПА С/П.

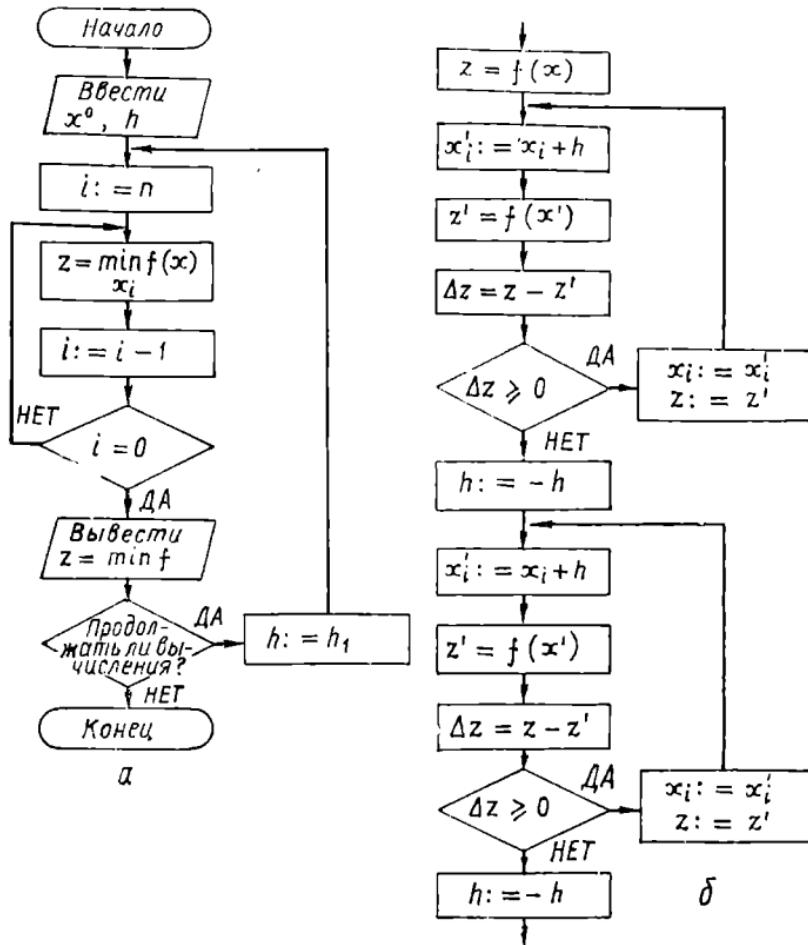


Рис. 23

Блок-схема вычислений по программе 36 представлена на рис. 23, а, подробная блок-схема поиска минимума по переменной x_i — на рис. 23, б.

Контрольный пример. Найти минимум функции $f(x, y) = x^2 + y + e^{-x-y}$ методом покоординатного спуска. В качестве начальной рассмотреть точку $M_0(0; 0)$, начальный шаг $h = 0,4$.

Подпрограмма вычисления $f(x, y)$:

ИП1 ИП2 + /—/ Fe^x ИП1 FX² + ИП2 + В/О

Нажмем клавиши F ПРГ и клавиши программы. Далее перейдем к автоматическому режиму F АВТ и введем начальные данные:

0 П1 П2 0,4 ПД.

Приступим к вычислениям: В/О С/П, через 25 с прочтем значение функции $f = 1$. Результаты следующих вычислений приводим в таблице:

Минимум по y		Минимум по x	
Клавиша С/П	Индикация 1 0.76	Клавиша С/П С/П	Индикация 0.83032004 0.76

Так как значения функции не убывают, уменьшим шаг, положив $h = 0.2; 0.2$ ПД ИПА

Клавиша С/П	Индикация 0.76	Клавиша С/П	Индикация 0.76
----------------	-------------------	----------------	-------------------

Далее выбираем шаг $h = 0.1: 0.1$ ПД ИПА

Клавиша С/П	Индикация 0.76	Клавиша С/П	Индикация 0.7548374
С/П	0.75	С/П	0.75

Для индикации значений x и y нажмем соответственно клавиши: ИП1 ($x = 0.5$) ИП2 ($y = -0.5$).

Ответ: минимум функции 0.75 достигается в точке $M(0.5; -0.5)$.

Легко убедиться в том, что для данного примера метод покоординатного спуска позволил найти точное значение минимума функции. Действительно, найдем частные производные функции и определим стационарные точки из системы (7.12):

$$\frac{\partial f}{\partial x} \equiv 2x - e^{-x-y} = 0, \quad \frac{\partial f}{\partial y} \equiv 1 - e^{-x-y} = 0.$$

Из второго уравнения $e^{-x-y} = 1$. Тогда из первого уравнения следует: $2x = 1$ и $x = 0.5$. Таким образом, $e^{-y} = e^x$ и $y = -0.5$. Точка $M(0.5; -0.5)$ будет стационарной точкой функции. Определим вторые производные функции

$$\frac{\partial^2 f}{\partial x^2} = 2 + e^{-x-y}; \quad \frac{\partial^2 f}{\partial x \partial y} = e^{-x-y}; \quad \frac{\partial^2 f}{\partial y^2} = e^{-x-y}$$

и составим матрицу Гессе в точке M

$$H(M) = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}.$$

Матрица $H(M)$ положительно определена, так как

$$a_{11} = 3 > 0 \quad \text{и} \quad \begin{vmatrix} 3 & 1 \\ 1 & 1 \end{vmatrix} = 2 > 0.$$

Вывод: точка $M(0.5; -0.5)$ является точкой минимума заданной функции и $f_{\min} = 0.75$.

Программа 37 вычисления минимума функции $f(x, y)$ методом градиентного спуска с постоянным шагом h по формулам (7.15) в автоматическом режиме:

Адресс	Команда	Код	Адресс	Команда	Код	Адресс	Команда	Код
00	ПП	53	11	ПП	53	22	00	00
01	23	23	12	(β)	β	23
02	С/П	50	13	ИП7	67
03	ПП	53	14	×	12	...	В/О	52
04	(α)	α	15	ИП2	62	α
05	ИП7	67	16	ХУ	14
06	×	12	17	—	11	...	В/О	52
07	ИП1	61	18	П2	42	β
08	ХУ	14	19	ИП3	63
09	—	11	20	П1	41	...	В/О	52
10	П3	43	21	БП	51			

Адресс (α) соответствует началу подпрограммы вычисления $f'_x(x, y)$, адресс (β) — началу подпрограммы вычисления $f'_y(x, y)$.

Распределение регистров памяти:

Номер регистра	1	2	3	7
Переменая	x	y	x	h

Инструкция. 1. Составить подпрограммы вычисления функции $f(x, y)$ и частных производных $f'_x(x, y)$ и $f'_y(x, y)$. Определить адресса (α) и (β).

2. Перейти в режим программирования, нажав клавиши F ПРГ. Ввести программу и подпрограмму.

3. Перейти в режим автоматической работы, нажав клавиши F АВТ.

4. Ввести начальные данные:

$$(x_0) \text{ П1 } (y_0) \text{ П2 } (h) \text{ П7}$$

5. Осуществить пуск программы: В/О С/П.

После каждого останова записать значение $f(x^k, y^k)$. Если нужно, записать x^k , нажав клавиши ИП1, и y^k , нажав клавиши ИП2. Нажать клавишу С/П.

Вычисления продолжаем до тех пор, пока последовательные значения $f(x, y)$ не станут достаточно близкими.

Контрольный пример. Найти минимум функции

$$f(x, y) = e^{y-x} + x^2 + y^2$$

методом градиентного спуска с постоянным шагом $h = 0,4$, выбрав начальную точку $M_0(1; 1)$. Составим подпрограммы вычисления $f(x, y)$, $f'_x(x, y) = -e^{y-x} + 2x$, $f'_y(x, y) = e^{y-x} + 2y$:

Адрес	23	24	25	26	27	28	29	30	31	32
Команда	ИП2	ИП1	—	Fe ^X	П5	ИП1	FX ²	ИП2	FX ²	+
Адрес	33	34	35	36	37	38	39	40	41	42
Команда	ИП5	+	В/О	ИП1	2	×	ИП5	—	В/О	ИП2
Адрес	43	44	45	46	47					
Команда	2	×	ИП5	+	В/О					

Адреса подпрограмм вычисления $f'_x(x, y)$ и $f'_y(x, y)$ равны соответственно $\alpha = 36$, $\beta = 42$.

Нажав клавиши F ПРГ, вводим программу и подпрограмму.

Нажав клавиши F АВТ, вводим исходные данные:

1 П1 П2 0,4 П7

Нажимаем клавиши В/О С/П. После останова спишем с индикатора $f(1; 1) = 3$. Нажимаем С/П, запишем $f(x^1, y^1) = 0,849$ и т. д. Результаты вычислений округляем до трех десятичных знаков и заносим в таблицу:

Номер шага k	0	1	2	3	4
$f(x^k, y^k)$	3	0,849	0,733	0,728	0,728

Если результаты начинают повторяться, то счет прекращается. Запишем координаты точки минимума:

ИП1 ($x = 0,284$); ИП2 ($y = -0,284$).

Ответ: $f_{min} = 0,728$; точка минимума $M(0,284; -0,284)$.

Задания к лабораторной работе 11. Найти минимум функции $f(x, y)$:

а) методом покоординатного спуска, выбрав начальную точку $M_0(x^0, y^0)$ и постоянный шаг h , в окрестности минимума уменьшить шаг, положив $h = 0,01$;

б) методом градиентного спуска. Поставить переключатель Р—Г в положение Р;

- 1) $f(x, y) = e^{x^2} + y + (x - y)^2$, $M_0(0; 0)$, $h = 0,1$;
- 2) $f(x, y) = e^{y^2-x} + e^x$, $M_0(-1; -1)$, $h = 0,1$;
- 3) $f(x, y) = e^{-x} + x^2 + y^2$, $M_0(1; 1)$, $h = 0,4$;
- 4) $f(x, y) = e^y + (y - x^2)^2$, $M_0(0,5; 0,5)$, $h = 0,1$;
- 5) $f(x, y) = e^{-y} - \cos(x^2 + y)$, $M_0(0; 0)$, $h = 0,4$;
- 6) $f(x, y) = e^x + y^2 - 2x$, $M_0(0; 0)$, $h = 0,4$;
- 7) $f(x, y) = x^3 - \cos(y - 1)$, $M_0(0; 0)$, $h = 0,4$;

- 8) $f(x, y) = y^2 + e^x - 3x$, $M_0(0; 0)$, $h = 0,4$;
 9) $f(x, y) = e^{x-y} + x^2 + y^2$, $M_0(1, 1)$, $h = 0,4$;
 10) $f(x, y) = e^{x^2-y} + e^y$, $M_0(-1; -1)$, $h = 0,4$;
 11) $f(x, y) = e^x + (x - y^2)^2$, $M_0(0,5; 0,5)$, $h = 0,1$;
 12) $f(x, y) = e^x + x^2 + y^2$, $M_0(-1; 1)$, $h = 0,4$;
 13) $f(x, y) = e^{y^2-x} + x^2$, $M_0(0; 0)$, $h = 0,4$;
 14) $f(x, y) = e^{-x} - \cos(x^2 + y^2)$, $M_0(0; 0)$, $h = 0,4$;
 15) $f(x, y) = e^{-y} + (y + x^2)^2$, $M_0(0,5; 0,5)$, $h = 0,1$;
 16) $f(x, y) = e^{x+y} + x^2 + y^2$, $M_0(-1; -1)$, $h = 0,4$;
 17) $f(x, y) = y^2 - \cos(x - 1)$, $M_0(0; 0)$, $h = 0,4$;
 18) $f(x, y) = x^2 + e^y - 3y$, $M_0(0; 0)$, $h = 0,4$;
 19) $f(x, y) = e^y + x^2 + y^2$, $M_0(1; 1)$, $h = 0,4$;
 20) $f(x, y) = e^{-x} + y^2 + 2x$, $M_0(0; 0)$, $h = 0,4$;
 21) $f(x, y) = e^{-y} + x^2 + y^2$, $M_0(1; -1)$, $h = 0,4$;
 22) $f(x, y) = e^x - \cos(x - y^2)$, $M_0(0; 0)$, $h = 0,4$;
 23) $f(x, y) = e^{-x} + (x + y^2)^2$, $M_0(0,5; 0,5)$, $h = 0,1$;
 24) $f(x, y) = e^y - \cos(x^2 - y)$, $M_0(0; 0)$, $h = 0,4$;
 25) $f(x, y) = e^{-x-y} + x^2 + y^2$, $M_0(1; 1)$, $h = 0,4$;
 26) $f(x, y) = x^2 - \cos(1 + y)$, $M_0(0; 0)$, $h = 0,4$;
 27) $f(x, y) = y^2 + e^{-x} + 3$, $M_0(0; 0)$, $h = 0,4$;
 28) $f(x, y) = e^y + x^2 - 2y$, $M_0(0; 0)$, $h = 0,4$;
 29) $f(x, y) = x^2 + e^{-y} + 3y$, $M_0(0, 0)$, $h = 0,4$;
 30) $f(x, y) = e^{-y} + x^2 + 2y$, $M_0(0, 0)$, $h = 0,4$.

Глава 8. ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ КОШИ ДЛЯ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

8.1. Метод Эйлера

Дифференциальные уравнения лежат в основе многих математических моделей, описывающих законы природы. Среди прикладных задач важное место занимает задача Коши, заключающаяся в нахождении частного решения $y = y(x)$ обыкновенного дифференциального уравнения первого порядка

$$y' = f(x, y), \quad (8.1)$$

удовлетворяющего заданному начальному условию

$$y(x_0) = y_0, \text{ где } M_0(x_0, y_0) \in D; \quad (8.2)$$

$$D = \{(x, y) | x_0 \leq x \leq x_0 + l, |y - y_0| \leq a\}.$$

График частного решения $y = y(x)$ есть интегральная кривая, проходящая через заданную в области D точку $M_0(x_0, y_0)$.

Известно, что решение задачи Коши (8.1), (8.2) на отрезке $[x_0; x_0 + h]$ существует и единствено, если функция $f(x, y)$ непрерывна в области D и удовлетворяет условию Липшица:

$$|f(x, y_1) - f(x, y_2)| \leq N |y_1 - y_2|, \quad (8.3)$$

где N — константа Липшица, $h = \min \{l, \frac{1}{N}, \frac{a}{M}\}$, $M = \max_D |f(x, y)|$. В частности, условие (8.3) выполняется, если $f(x, y)$ имеет в D ограниченную частную производную $f'_y(x, y)$.

Поскольку точное решение задачи Коши возможно лишь в исключительно редких случаях, то для нахождения частного решения $y(x)$ в основном применяют численные методы. В области D на выбранной последовательности точек $x_0, x_1, x_2, \dots, x_n, x_i \in [x_0, x_0 + l]$ с переменной в общем случае длиной шага $h_i = x_{i+1} - x_i$ вычисляются приближенные значения $y_0, y_1, y_2, \dots, y_n$ искомого решения задачи Коши.

Большинство численных методов можно представить в виде зависимости

$$y_{i+1} = \varphi(y_{i-r}, y_{i-r+1}, \dots, y_i, y_{i+1}, \dots, y_{i+s}).$$

Функция φ определяет вычислительную схему метода. Если $r = 0$ и $0 < s \leq 1$, то численный метод называется *одношаговым*, а если $r \geq 1$ или $s > 1$, то *многошаговым*.

Многошаговые и одношаговые методы называются *явными*, если $s = 0$, и *неявными* при $s = 1$. При $s > 1$ многошаговые методы называются методами с забеганием вперед.

Многие численные методы, разработанные для решения одного дифференциального уравнения, легко обобщаются на решение систем обыкновенных дифференциальных уравнений первого порядка.

Задачи Коши для нормальной системы обыкновенных дифференциальных уравнений

$$\begin{aligned} \frac{dy_1}{dx} &= f_1(x, y_1, \dots, y_n); \\ \frac{dy_2}{dx} &= f_2(x, y_1, \dots, y_n); \\ \frac{dy_n}{dx} &= f_n(x, y_1, \dots, y_n) \end{aligned} \quad (8.4)$$

состоит в нахождении частного решения $y_1(x), y_2(x), \dots, y_n(x)$, удовлетворяющего начальным условиям

$$y_1(x_0) = y_{10}, y_2(x_0) = y_{20}, \dots, y_n(x_0) = y_{n0}. \quad (8.5)$$

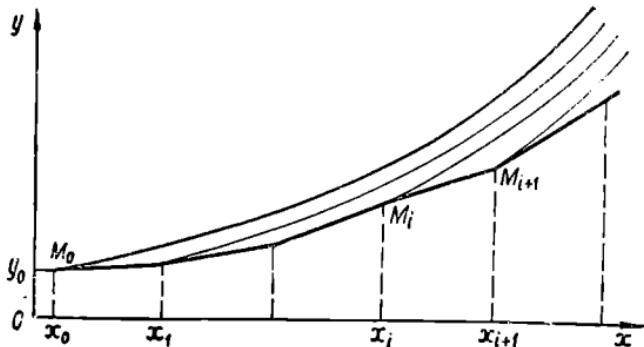


Рис. 24

Задача Коши для обыкновенного дифференциального уравнения n -го порядка

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)})$$

и начальных условий

$$y(x_0) = y_0, y'(x_0) = y'_0, \dots, y^{(n-1)}(x_0) = y_0^{n-1}$$

может быть сведена к решению задачи Коши для системы дифференциальных уравнений. Поэтому в дальнейшем будут излагаться численные методы решения задачи Коши для одного уравнения (8.1), (8.2).

Метод Эйлера (или метод ломанных) является простейшим одношаговым численным методом решения задачи Коши (8.1), (8.2). В методе Эйлера приближенные значения искомого частного решения $y(x_i) = y_i$ на отрезке $[x_0, x_0 + l]$ вычисляются последовательно по формуле

$$y_{i+1} = y_i + hf(x_i, y_i); \quad (i = 0, 1, 2, \dots, n-1). \quad (8.6)$$

Здесь шаг интегрирования h предполагается постоянным, $h = \frac{l}{n}$.

Геометрическая интерпретация метода Эйлера следующая (рис. 24): на каждом i -м шаге, начиная с точки $M_0(x_0, y_0)$, движение осуществляется не по интегральной кривой, а по отрезку касательной M_iM_{i+1} к интегральной кривой, проходящей через точку $M_i(x_i, y_i)$. Уравнение такой касательной

$$y = y_i + y'(x_i)(x - x_i),$$

а так как $y'(x_i) = f(x_i, y_i)$ и $x_{i+1} = x_i + h$, то

$$y_{i+1} = y_i + hf(x_i, y_i).$$

Для оценки локальной погрешности метода на одном шаге разложим искомое решение $y(x)$ в ряд Тейлора на интервале $[x_i; x_{i+1}]$:

$$y_{i+1} = y_i + hy'_i + \frac{1}{2}h^2y''_i + \dots \quad (8.7)$$

Производные $y_i^{(k)}$ можно найти, дифференцируя исходные уравнения

$$y' = f(x, y); \quad y'' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \cdot f(x, y), \text{ и т. д.}$$

Удерживая в (8.7) члены до $O(h^p)$, получим численный метод порядка p .

Метод Эйлера имеет первый порядок точности. С удалением от начальной точки (x_0, y_0) погрешности в значениях y_i накапливаются.

Метод достаточно прост и нагляден, чтобы изложить на его примере идею построения более точных методов численного интегрирования дифференциальных уравнений.

В практических вычислениях чаще применяются модификации метода Эйлера. Так, в усовершенствованном методе Эйлера—Коши сначала на $(i+1)$ -м шаге определяется приближение как и в обычном методе Эйлера (8.6):

$$y_{i+1}^* = y_i + hf(x_i, y_i), \quad (8.8)$$

в найденной точке вычисляется направление поля

$$f(x_{i+1}, y_{i+1}^*),$$

а затем вычисления проводятся по формуле

$$y_{i+1} = y_i + h \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^*)}{2}. \quad (8.9)$$

В вычислительной практике применяется также метод Эйлера—Коши с итерациями: по грубому первоначальному приближению

$$y_{i+1}^{(0)} = y_i + hf(x_i, y_i) \quad (8.10)$$

строится итерационный процесс

$$y_{i+1}^{(k)} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k-1)})], \quad k = 1, 2, 3, \dots. \quad (8.11)$$

Итерации продолжают до тех пор, пока два последовательных приближения не совпадут с заданной точностью ε . Если после нескольких итераций совпадение не наблюдается, то нужно уменьшить шаг.

Указанные модификации метода Эйлера (8.8), (8.9) и (8.10), (8.11) называют еще методами прогноза и коррекции (предиктор-корректор). Здесь на первом этапе, называемом прогнозом, по формулам (8.8) или (8.10) находится приближение для y_{i+1} , а затем на втором этапе оно корректируется по формулам (8.9) или (8.11). Обе модификации являются методами второго порядка точности.

Абсолютная локальная погрешность этих методов на каждом шаге пропорциональна h^3 .

Метод Эйлера и его модификации обобщаются на численное решение задачи Коши для систем обыкновенных дифференциальных уравнений (8.4) и (8.5).

Программа 38 приближенного решения задачи Коши (8.1), (8.2) для дифференциального уравнения первого порядка методом Эйлера—Коши с итерациями (8.10), (8.11).

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	ПП	53	14	+	10	28	FX ²	22
01	38	38	15	П1	41	29	ИП4	64
02	ИП3	63	16	ПП	53	30	—	11
03	×	12	17	38	38	31	FX < 0	5□
04	2	02	18	ИП3	63	32	16	16
05	÷	13	19	×	12	33	ИП2	62
06	↑	0E	20	2	02	34	C/П	50
07	ИП2	62	21	÷	13	35	FL0	5□
08	+	10	22	ИП5	65	36	00	00
09	П5	45	23	+	10	37	C/П	50
10	+	10	24	ИП2	62	38
11	П2	42	25	ХУ	14
12	ИП1	61	26	П2	42
13	ИП3	63	27	—	11	...	В/О	52

Распределение регистров памяти:

Номер регистра	0	1	2	3	4
Переменная	n , счетчик цикла	x_0, x_i	y_0, y_i	$h = \frac{l}{n}$	ε^2

Инструкция. 1. Составить подпрограмму вычисления значений $f(x, y)$, значение x хранится в Р1, значение y — в Р2.

2. Нажать клавиши F ПРГ, ввести программу и подпрограмму.

3. Нажать клавиши F АВТ. Ввести исходные данные:

(n) ПО, (x_0) П1, (y_0) П2, (h) П3, (ε^2) П4.

4. Нажать клавиши В/О С/П. После каждого ОСТАНОВа записать значение y_{i+1} и нажать клавишу С/П.

Контрольный пример. Методом Эйлера—Коши с итерациями найти приближенное решение уравнения $y' = x - y$, удовлетворяющее начальному условию $y(1) = -1$, на отрезке $[1; 1,6]$. Выбрать шаг интегрирования $h = 0,2$. Итерации продолжать до совпадения последовательных приближений с точностью $\varepsilon = 10^{-2}$.

Результаты вычислений округлены до трех десятичных знаков и выписаны в таблице. Ниже для сравнения приведено точное значение решения задачи Коши (с тремя десятичными знаками), которое в данном случае равно $y(x) = -(e^{1-x} - 1 - x)$.

x_i	1	1,2	1,4	1,6
y_i^k	-1	-0,618	-0,269	0,053
$y_i^{\text{точн}}$	-1	-0,619	-0,270	0,051

8.2. Метод Рунге–Кутта

Метод Рунге–Кутта позволяет строить схемы различного порядка точности и наиболее применим в практических вычислениях. Рассмотрим вначале метод второго порядка, который требует лишь два вычисления функции $f(x, y)$ на каждом шаге. Одно из них равно

$$k_1 = h_i f(x_i, y_i) = h_i f_i.$$

Затем осуществляется дробный шаг, использующий k_1 ,

$$k_2 = h_i f(x_i + \alpha h_i, y_i + \beta k_1),$$

где α и β — неизвестные числа.

Для полного шага берется комбинация значений функции

$$y_{i+1} = y_i + \gamma_1 k_1 + \gamma_2 k_2;$$

γ_1 и γ_2 подлежат определению.

Для определения неизвестных коэффициентов разложим k_1 и k_2 в ряд Тейлора в окрестности точки (x_i, y_i) :

$$k_1 = h_i \cdot f_i, \quad k_2 = h_i \left(f_i + \alpha h_i \cdot \frac{\partial f_i}{\partial x} + \beta k_1 \frac{\partial f_i}{\partial y} + \dots \right),$$

$$y_{i+1} = y_i + (\gamma_1 + \gamma_2) h_i f_i + \gamma_2 \beta h_i^2 \frac{\partial f_i}{\partial y} f_i + \gamma_2 \alpha h_i^2 \cdot \frac{\partial f_i}{\partial x} \dots$$

Сравнив это разложение с разложением (§.7) точного решения задачи Коши (8.1), (8.2) на i -м шаге, получим уравнения:

$$\gamma_1 + \gamma_2 = 1; \quad \gamma_2 \beta = \frac{1}{2}; \quad \gamma_2 \alpha = \frac{1}{2}$$

(α можно выбрать произвольно, положив его равным $\frac{1}{2}$ или 1). Ни при каких α не удается исключить члены, содержащие h_i^3 , таким образом, метод имеет второй порядок точности.

При $\alpha = \frac{1}{2}$ получаем вычислительную схему;

$$y_{i+1} = y_i + h_i f \left(x_i + \frac{1}{2} h_i; \quad y_i + \frac{1}{2} h_i f_i \right). \quad (8.13)$$

Геометрическая интерпретация формулы (8.13) следующая: вначале движемся на половинном шаге по ломаной Эйлера $y_{i+\frac{1}{2}} = y_i + \frac{h_i}{2} f_i$, затем в найденной точке вычисляем наклон интегральной кривой $y'_{i+\frac{1}{2}} = f \left(x_i + \frac{1}{2} h_i, \quad y_{i+\frac{1}{2}} \right)$. По этому наклону определяем приращение функции на целом шаге:

$$\frac{y_{i+1} - y_i}{h_i} = y'_{i+\frac{1}{2}}.$$

При $\alpha = 1$ получаем формулу

$$y_{i+1} = y_i + \frac{h_i}{2} [f(x_i, y_i) + f(x_i + h_i, y_i + h_i f_i)], \quad (8.14)$$

совпадающую с формулами (8.8) и (8.9) усовершенствованного метода Эйлера — Коши.

Аналогично строятся вычислительные схемы метода Рунге — Кутта более высоких порядков.

Наиболее употребительны схемы четвертого порядка точности. В частности, в большинстве стандартных программ ЭВМ используется формула

$$y_{i+1} = y_i + \frac{h_i}{6} (k_1 + 2k_2 + 2k_3 + k_4), \quad (8.15)$$

где $k_1 = f(x_i, y_i)$; $k_2 = f \left(x_i + \frac{h_i}{2}, y_i + \frac{h_i}{2} k_1 \right)$,

$$k_3 = f \left(x_i + \frac{h_i}{2}; \quad y_i + \frac{h_i}{2} k_2 \right); \quad k_4 = f(x_i + h_i, y_i + h_i k_3).$$

Методы Рунге—Кутта легко программировать, они численно устойчивы для широкого класса задач, величину шага можно изменять. Эти методы легко переносятся на решение задачи Коши для системы дифференциальных уравнений.

Предельная абсолютная локальная погрешность метода Рунге—Кутта (8.15) на каждом шаге пропорциональна h^5 .

Точная оценка погрешности методов численного интегрирования громоздка. Для практической оценки погрешности используют правило Рунге: чтобы приближенно оценить погрешность метода численного интегрирования p -го порядка, необходимо провести двойной пересчет значения $y(x_i)$ с шагом h и с шагом $h/2$, а затем полагают

$$|y(x_i) - y_{h/2}(x_i)| \approx \frac{|y_h(x_i) - y_{h/2}(x_i)|}{2^p - 1}, \quad (8.16)$$

где $y(x_i)$, $y_h(x_i)$, $y_{h/2}(x_i)$ — соответственно точное решение уравнения и приближенные решения, вычисленные в одной и той же точке x_i ($i = \overline{1, n}$) с шагом h и с шагом $h/2$; p — порядок метода.

Для усовершенствованного метода Эйлера — Коши и этого же метода с итерациями оценка (8.16) приобретает вид

$$|y(x_i) - y_{h/2}(x_i)| \approx \frac{|y_h(x_i) - y_{h/2}(x_i)|}{3}, \quad (8.17)$$

а для метода Рунге — Кутта (8.15) четвертого порядка точности имеем

$$|y(x_i) - y_{h/2}(x_i)| \approx \frac{|y_h(x_i) - y_{h/2}(x_i)|}{15}. \quad (8.18)$$

Проверку точности вычислений с помощью двойного пересчета достаточно выполнять через 4—5 шагов. Выбрав условие

$$\delta = \frac{|y_h(x_i) - y_{h/2}(x_i)|}{15} < \epsilon \quad (8.19)$$

в качестве критерия выбора шага для получения решения с заданной точностью ϵ , можно составить программу численного интегрирования по методу Рунге—Кутта (8.15) с автоматическим выбором шага.

8.3. Метод Адамса

В рассмотренных выше одношаговых методах численного интегрирования дифференциальных уравнений используют на каждом шаге вычислений информацию о значениях искомого

решения только в предыдущей точке. Вычислительные схемы многошаговых методов строятся так, что полученная ранее информация о решении используется повторно на нескольких шагах.

Среди многошаговых методов различных порядков наиболее употребительны методы Адамса, разработанные им в 1855 г. Идея методов состоит в следующем: пусть найдены приближенные значения $y(x)$ в $k+1$ точках $x_i, x_{i-1}, \dots, x_{i-k}$ с постоянным шагом h . Обозначим $y(x_i) = y_i; f(x_i, y_i) = f_i$. По данным узлам строим интерполяционный полином $P_k(x)$ для функции $f(x, y(x))$. Считая приближенно $y(x) = P_k(x)$, проинтегрируем это равенство на интервале $[x_i; x_{i+1}]$:

$$y_{i+1} \approx y_i + \int_{x_i}^{x_{i+1}} P_k(x) dx. \quad (8.20)$$

$P_k(x)$ можно выразить через конечные разности функции $f(x, y(x))$, используя формулу Ньютона для интерполирования назад (5.16). Например, если ограничиться разностями третьего порядка, то получим следующую экстраполяционную формулу Адамса

$$y_{i+1} = y_i + F_i + \frac{1}{2} \Delta F_{i-1} + \frac{5}{12} \Delta^2 F_{i-2} + \frac{3}{8} \Delta^3 F_{i-3}, \quad (8.21)$$

где $F_i = hf(x_i, y(x_i))$.

При использовании вычислительной техники (8.21) удобнее представить в другой записи. Выразив конечные разности через значения функции

$$\begin{aligned} \Delta F_{i-1} &= h(f_i - f_{i-1}); \quad \Delta^2 F_{i-2} = h(f_i - 2f_{i-1} + f_{i-2}); \\ \Delta^3 F_{i-3} &= h(f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}) \end{aligned}$$

и подставив их в (8.21), получим экстраполяционную формулу Адамса — Башфорта

$$y_{i+1} = y_i + \frac{h}{24} (55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}). \quad (8.22)$$

Здесь используются значения функции f_i в точках $x_i, x_{i-1}, x_{i-2}, x_{i-3}$, предшествующих отрезку интегрирования. Такое приближение называется экстраполяцией.

После того как вычислено значение y_{i+1} по (8.21), определяют $f_{i+1} = f(x_{i+1}, y_{i+1})$, а затем уточняют y_{i+1} по следующей интерполяционной формуле Адамса — Мултона:

$$y_{i+1} = y_i + \frac{h}{24} (9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}). \quad (8.23)$$

Сна может быть получена аналогично формуле (8.21) из (8.20), если вместо $P_k(x)$ подставить полином Ньютона для интерполяции назад (5.16), построенный по точкам $x_{i+1}, x_i, x_{i-1}, x_{i-2}$.

Формулы (8.22) и (8.23) имеют четвертый порядок точности. Для начала вычислений нужны, кроме y_0 , еще три значения искомого решения y_1, y_2, y_3 . Обычно их определяют каким-либо одношаговым методом того же порядка, например, методом Рунге—Кутта. Величину шага в методе Рунге—Кутта выбирают меньшей, чем в последующих вычислениях, для достижения большей точности в начальных точках отрезка интегрирования.

Разность между предсказанным значением $y_{i+1}^{\text{пред}}$ и скорректированным $y_{i+1}^{\text{кор.}}$, полученными соответственно по формулам (8.22) и (8.23), можно принять приближенно в качестве оценки локальной погрешности вычислений на данном шаге. При подходящем выборе шага h ее можно сделать меньше наперед заданного ε .

Важное преимущество многошаговых методов — меньший объем вычислений функции $f(x, y)$ на каждом шаге. Кроме того, они допускают экономный автоматический контроль шага по разности $|y_{i+1}^{\text{кор.}} - y_{i+1}^{\text{пред.}}|$. К их неудобствам следует отнести проблему построения начала таблицы.

Методы Адамса можно использовать для приближенного решения задачи Коши (8.4), (8.5) для системы дифференциальных уравнений.

8.4. Метод Милна

Метод Милна является многошаговым методом четвертого порядка точности типа «предсказание — коррекция». Для его старта необходимо найти каким-либо одношаговым методом четыре значения искомого решения y_0, y_1, y_2, y_3 . Дальнейшие вычисления проводятся по схеме:

1. По четырем предыдущим точкам предсказываем следующее значение y_{i+1} :

$$y_{i+1}^{\text{пред.}} = y_{i-3} + \frac{4}{3} h (2f_i - f_{i-1} + 2f_{i-2}), \quad (8.24)$$

где $f_i = f(x_i, y_i)$; ($i = 3, 4, 5, \dots$).

2. Вычисляем значения правой части уравнения

$$f_{i+1}^{\text{пред.}} = f(x_{i+1}, y_{i+1}^{\text{пред.}}).$$

3. Корректируем значение y_{i+1}

$$y_{i+1}^{\text{кор.}} = y_{i-1} + \frac{h}{3} (f_{i-1} + 4f_i + f_{i+1}^{\text{пред.}}), \quad (i = 3, 4, \dots). \quad (8.25)$$

Пределная абсолютная погрешность значения y_i в методе Милна равна

$$\epsilon_i = \frac{1}{29} |y_i^{\text{пред.}} - y_i^{\text{кор.}}|. \quad (8.26)$$

8.5. Лабораторная работа 12. Численные методы решения задачи Коши

Решение задачи Коши (8.1), (8.2) для обыкновенного дифференциального уравнения первого порядка методом Рунге—Кутта четвертого порядка с постоянным шагом h реализуется на микрокалькуляторе с помощью программы 39.

Программа 39 решения задачи Коши (8.1), (8.2) методом Рунге—Кутта четвертого порядка по формулам (8.15) с постоянным шагом h в автоматическом режиме

Адресс	Команд-да	Код	Адресс	Коман-да	Код	Адресс	Коман-да	Код
00	C_x	0Г	16	ПП	53	32	+	10
01	П5	45	17	26	26	33	П5	45
02	ИП4	64	18	ИП5	65	34	ИП4	64
03	П2	42	19	+	10	35	ИП6	66
04	ПП	53	20	3	03	36	+	10
05	38	38	21	÷	13	37	П2	42
06	ПП	53	22	ИП4	64	38	ПП	53
07	26	26	23	+	10	39	44	44
08	ИП5	65	24	П4	44	40	ИП3	63
09	+	10	25	С/П	50	41	×	12
10	П5	45	26	ИП1	61	42	П6	46
11	ПП	53	27	ИП3	63	43	В/О	52
12	30	30	28	+	10	44
13	2	02	29	П1	41	$f(x, y)$
14	×	12	30	ИП6	66	...	В/О	52
15	П6	46	31	ИП5	65			

Распределение регистров памяти:

Номер регистра	1	2	3	4
Переменная	x_i, x	y	$h/2$	Δt

Инструкция. 1. Составить подпрограмму вычисления значения функции $f(x, y)$. Значения x и y хранятся соответственно в регистрах 1 и 2.

2. В режиме программирования (F ПРГ) ввести программу и подпрограмму.

3. В автоматическом режиме (F АВТ) ввести исходные данные (x_0) П1 (y_0) П4 $(h/2)$ П3.

4. Нажать клавиши В/О С/П. После останова записать с индикатора значение $y_1 = y(x_0 + h)$.

5. Для вычисления каждого следующего значения $y_i = y(x_0 + ih)$ ($i = 2, 3, \dots, n$) повторить п. 4.

При составлении подпрограммы вычисления $f(x, y)$ можно использовать 54 адреса программной памяти и свободные регистры 7, 8, 9, A, B, C, D.

Для контроля точности можно провести вычисления с шагом $h/2$ и воспользоваться оценкой Рунге (8.18). Для этого необходимо ввести исходные данные (x_0) П1 (y_0) П4 $(h/4)$ П3 и повторить вычисления.

Контрольный пример. Найти приближенное решение дифференциального уравнения $y' = y^2 + x^3$, удовлетворяющее начальным условиям $y(0) = 0$, на отрезке $[0; 1]$ с шагом $h = 0,2$.

Подпрограмма вычисления функции $f(x; y) = y^2 + x^3$: ИП2 FX² ИП1 ↑ FX² × + В/О. Вводим исходные данные: 0 П1 П4 0,1 П3 В/О С/П. После ОСТАНОВа (примерно через 30 с) прочтем на индикаторе $y_1 = y(0,2) = 0,0004000$.

Результаты вычислений удобно заносить в таблицу:

x_i	0,2	0,4	0,6	0,8	1,0
y_i, h	0,0004000	0,0064019	0,0324710	0,1033483	0,2572235
$y_i, h/2$	0,0004000	0,0064018	0,0324703	0,1033436	0,2572037

Для получения следующего значения y_i нажимаем клавиши В/О С/П и заполняем таблицу. Повторим вычисления с шагом $h = 0,1$. Введем исходные данные 0 П1 П4 0,05 П3 В/О. Вычисленные повторно с шагом $h/2$ значения y_i в точках $x = 0,2; 0,4; 0,6; 0,8; 1,0$ (через одно) запишем в таблицу. Из результатов вычислений следует, что расхождение между y_i, h и $y_i, h/2$ с удалением от точки x_0 возрастает. Оценим наибольшую погрешность в точке $x = 1,0$:

$$\delta = \frac{|y_{5,h} - y_{5,h/2}|}{15} = 1,32 \cdot 10^{-6} < 10^{-5}.$$

Таким образом, во всех точках x_i отрезка $[0; 1]$ значения частного решения задачи Коши $y(x_i)$ получены с пятью верными десятичными знаками.

Выбор шага по условию (8.19) для получения решения с заданной точностью ϵ можно запрограммировать, увеличивая или уменьшая шаг, в зависимости от величины δ . Такие вычисления реализуются программой 40.

Программа 40 решения задачи Коши для дифференциального уравнения первого порядка (8.1), (8.2) методом Рунге—Кутта (8.15) с переменным шагом (в полуавтоматическом режиме). Блок-схема программы представлена на рис. 25.

Адрес	Команды	Код	Адрес	Команды	Код	Адрес	Команда	Код
00	ИП7	67	27	БП	51	54	÷	13
01	П3	43	28	04	04	55	ИП4	64
02	ПП	53	29	ИПА	6—	56	+	10
03	29	29	30	П!	41	57	В/О	52
04	ПД	4Г	31	ИПВ	6Л	58	ИП1	61
05	ИП7	67	32	П4	44	59	ИП3	63
06	2	02	33	0	00	60	+	10
07	÷	13	34	П5	45	61	П1	41
08	П3	43	35	ИП4	64	62	ИП6	66
09	ПП	53	36	П2	42	63	ИП5	65
10	29	29	37	ПП	53	64	+	10
11	П4	44	38	70	70	65	П5	45
12	ПП	53	39	ПП	53	66	ИП4	64
13	33	33	40	58	58	67	ИП6	66
14	П4	44	41	ИП5	65	68	+	10
15	ИПД	6Г	42	+	10	69	П2	42
16	—	11	43	П5	45	70	ПП	53
17	1	01	44	ПП	53	71	76	76
18	5	05	45	62	62	72	ИП3	63
19	÷	13	46	2	02	73	×	12
20	С/П	50	47	×	12	74	П6	46
21	ИП1	61	48	П6	46	75	В/О	52
22	ПА	4—	49	ПП	53	76	... } f(x, y)	...
23	ИП4	64	50	58	58
24	ПВ	4L	51	ИП5	65	...	В/О	52
25	ПП	53	52	+	10			
26	33	33	53	3	03			

Распределение регистров памяти:

Номер регистра	1	2	3,7	4	A	B
Переменная	x	y	$h/2$	$y_i, h/2$	x_i	y_i

Инструкция. 1. Составить подпрограмму вычисления $f(x, y)$, значения x и y хранятся соответственно в регистрах 1 и 2. При этом можно использовать 22 адреса программной памяти и свободные регистры 0, 8, С.

2. F ПРГ. Ввести программу и подпрограмму.

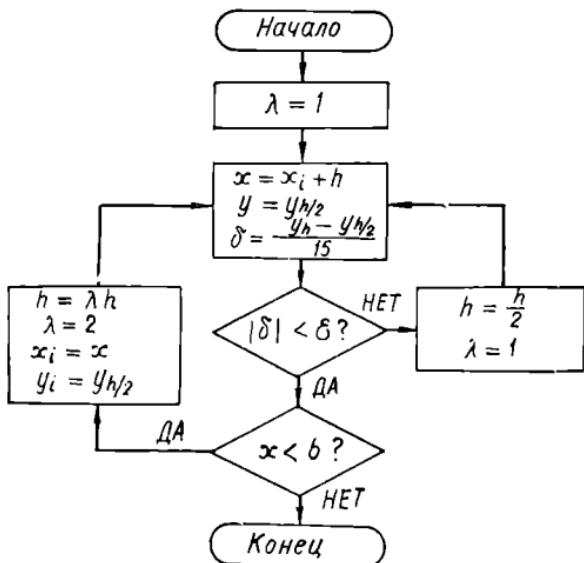


Рис. 25

3. F АВТ. Ввести исходные данные: (x_0) ПА (y_0) ПВ $(h/2)$ П7.

4. После нажатия клавиш В/О С/П производятся следующие вычисления. Методом Рунге—Кутта находится $y_{1,h}$ с шагом h и $y_{1,h/2}$ с шагом $h/2$, вычисляется по формуле (8.19) приближенная погрешность $\delta = (y_{1,h} - y_{1,h/2}) / 15$ одного шага и высвечивается на индикаторе. Значение $y_{1,h/2}$ хранится в Р4. Если $|\delta| < \epsilon$, продолжаем вычисления с тем же шагом, нажав клавишу С/П. Для продолжения вычислений с другим шагом h_1 ($h_1 = 2h$ или $h_1 = h/2$) следует ввести новое значение шага в Р7 ($h_1/2$) П7 и нажать клавишу С/П. Для повторного вычисления с шагом h_1 следует ввести значение $h_1/2$ в Р7 и нажать клавиши В/О С/П.

Контрольный пример. Найти частное решение задачи Коши $y' = xy^2 + \frac{1}{1-x}$, $y(0) = 1$ на отрезке $[0; 0,4]$ методом Рунге—Кутта с переменным шагом с точностью $\epsilon = 10^{-8}$. Выбрать начальный шаг $h = 0,2$.

Составим подпрограмму вычисления $f(x, y)$:

ИП1 ИП2 FX² × 1 ИП1 — F1/X + В/О

Вводим начальные данные:

0 ПА 1 ПВ 0,1 П7

Вычисления проводились в такой последовательности:

Нажимаемые клавиши	Индикация
B/O C/P	$\delta = 2 \cdot 10^{-8}$
ИП14	$y(0,2) = 1,250000$
C/P	$\delta = 1,6 \cdot 10^{-2}$
0,05 П7 B/O C/P	$\delta = -3,3 \cdot 10^{-8}$
ИП14	$y(0,3) = 1,428572$
C/P	$\delta = 8,5 \cdot 10^{-3}$
0,025 П7 B/O C/P	$\delta = 0$
ИП14	$y(0,35) = 1,538462$
C/P	$\delta = 4,4 \cdot 10^{-3}$
0,0125 П7 B/O C/P	$\delta = 0$
ИП14	$y(0,375) = 1,6000000$
C/P	$\delta = 0$
ИП4	$y(0,4) = 1,666667$

Численные значения y_t округляем до шести десятичных знаков и заносим в таблицу:

x_t	0	0,2	0,3	0,35	0,375	0,4
y_t	1	1,250000	1,428572	1,538462	1,600000	1,666667

Время вычислений на одном шаге составляет около 2 мин.

Задания к лабораторной работе 12.

- Найти численное решение задачи Коши для заданных дифференциальных уравнений и начальных условий методом Рунге—Кутта (8.15) с постоянным шагом $h = 0,02$ на отрезке $[0; 0,06]$.
- Повторить вычисления с шагом $h = 0,01$ и оценить погрешность по правилу Рунге (8.18).
- Найти решение задачи Коши методом Рунге—Кутта с переменным шагом с точностью $\epsilon = 10^{-6}$. Выбрать начальный шаг $h = 0,02$. Переключатель Р—Г поставить в положение Р.

- | | |
|--|--|
| 1. $y' = x^2 + y^2$, $y(0) = 1$; | 13. $y' = y^2 - x^2$, $y(0) = 0$; |
| 2. $y' = x + y^2$, $y(0) = 1$; | 14. $y' = x - y^2$, $y(0) = 1$; |
| 3. $y' = \cos x + y^2$, $y(0) = 0$; | 15. $y' = \cos x - y^2$, $y(0) = 0$; |
| 4. $y' = e^x + y^2$, $y(0) = 0$; | 16. $y' = e^x - y^2$, $y(0) = 0$; |
| 5. $y' = x + e^y$, $y(0) = 0$; | 17. $y' = x - e^y$, $y(0) = 0$; |
| 6. $y' = x + \cos y$, $y(0) = 0$; | 18. $y' = x - \cos y$, $y(0) = 0$; |
| 7. $y' = x^2 + e^y$, $y(0) = 0$; | 19. $y' = x^2 - e^y$, $y(0) = 0$; |
| 8. $y' = x + \sin y$, $y(0) = 1$; | 20. $y' = x - \sin y$, $y(0) = 1$; |
| 9. $y' = \sin x + y^2$, $y(0) = 1$; | 21. $y' = \sin x - y^2$, $y(0) = 1$; |
| 10. $y' = \sin x + e^y$, $y(0) = 0$; | 22. $y' = \sin x - e^y$, $y(0) = 0$; |
| 11. $y' = e^x + \sin y$, $y(0) = 0$; | 23. $y' = e^x - \sin y$, $y(0) = 0$; |
| 12. $y' = x^2 - y^2$, $y(0) = 1$; | 24. $y' = \sin(x + y)$, $y(0) = 1$; |

25. $y' = x + \ln(x + y)$, $y(0) = 3$; 28. $y' = x - e^y$, $y(0) = 1$;
 26. $y' = x + y^3$, $y(0) = 1$; 29. $y' = y + xe^y$, $y(0) = 1$;
 27. $y' = x - y^2$, $y(0) = 1$; 30. $y' = y^2 + \frac{x}{y}$, $y(0) = 1$.

Глава 9. РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ МЕТОДОМ СЕТОК

9.1. Метод сеток

Дифференциальные уравнения в частных производных моделируют разнообразные физические процессы, в которых исследуемая функция зависит от нескольких переменных. Метод сеток, или метод конечных разностей, является наиболее распространенным и эффективным методом численного решения краевых задач для уравнений математической физики. Идея этого метода принадлежит Эйлеру. Сущность его состоит в следующем. Область D непрерывного изменения аргументов в исходной задаче заменяется конечным (дискретным) множеством точек (узлов) G , называемых сеткой. Дифференциальное уравнение заменяется конечно-разностным уравнением, при этом производные искомой функции в выбранных узлах заменяются разделенными разностями. Дифференциальное уравнение заменяется системой алгебраических уравнений для функций дискретных аргументов (сеточной функции), определенной в узлах сетки. Начальные и краевые условия заменяются разностными начальными и краевыми условиями для сеточной функции. Полученную систему конечно-разностных (алгебраических) уравнений решаем каким-либо методом и определяем значения искомой функции в узлах сетки, т. е. численное решение исходной задачи.

Сетка может быть построена по-разному с учетом конкретных условий решаемой задачи. На плоскости часто применяют квадратные, прямоугольные, треугольные, шестиугольные и другие сетки. Ограничимся рассмотрением прямоугольной сетки, образованной системой прямых, $x_i = x_0 + ih$, $y_k = y_0 + kl$, ($i, k = 0, \pm 1, \pm 2, \dots$), постоянные положительные числа h и l называются шагом сетки по оси OX и OY соответственно. Точки (x_i, y_k) пересечения прямых называются узлами сетки. Два узла называются соседними, если они удалены друг от друга в направлении осей OX или OY на расстояние, равное шагу по этой оси. Узлы сетки G или принадлежат области D , или отстоят от ее границы Γ на расстоянии, меньшем шага. Узел сетки называется внутренним, если все его четыре соседних узла принадлежат области D .

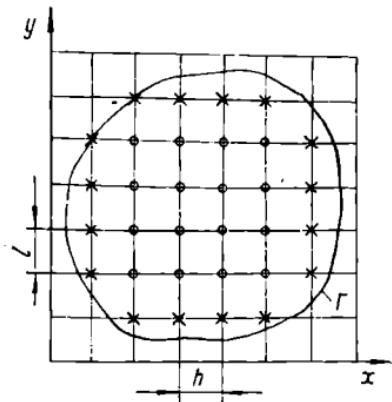


Рис. 26

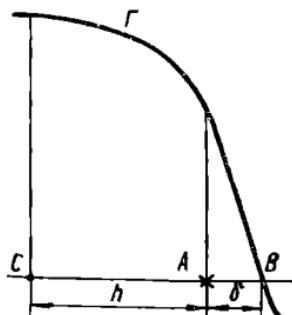


Рис. 27

На рис. 26 внутренние узлы сетки обозначены кружочками, а граничные — звездочками.

Аппроксимация дифференциального уравнения разностным. Обозначим значение искомой функции $u(x, y)$ в узле (x_i, y_k) через $u_{i,k}$. В каждом внутреннем узле сетки заменим частные производные функции их конечными разностями, используя формулы численного дифференцирования:

$$\frac{\partial u(x_i, y_k)}{\partial x} \approx \frac{u_{i+1,k} - u_{i-1,k}}{2h}; \quad \frac{\partial u(x_i, y_k)}{\partial y} \approx \frac{u_{i,k+1} - u_{i,k-1}}{2l}. \quad (9.1)$$

В граничных точках применяются менее точные разности

$$\frac{\partial u(x_i, y_k)}{\partial x} \approx \frac{u_{i+1,k} - u_{i,k}}{h}; \quad \frac{\partial u(x_i, y_k)}{\partial y} \approx \frac{u_{i,k+1} - u_{i,k}}{l}. \quad (9.2)$$

Вторые частные производные в каждом внутреннем узле заменяются разностями

$$\begin{aligned} \frac{\partial^2 u(x_i, y_k)}{\partial x^2} &\approx \frac{u_{i+1,k} - 2u_{i,k} + u_{i-1,k}}{h^2}; \\ \frac{\partial^2 u(x_i, y_k)}{\partial y^2} &\approx \frac{u_{i,k+1} - 2u_{i,k} + u_{i,k-1}}{l^2}; \\ \frac{\partial^2 u(x_i, y_k)}{\partial x \partial y} &\approx \frac{u_{i+1,k+1} - u_{i+1,k-1} - u_{i-1,k+1} + u_{i-1,k-1}}{4hl}. \end{aligned} \quad (9.3)$$

В результате такой замены в каждом внутреннем узле $(x_i, y_k) \in G$ получим конечно-разностное (алгебраическое) уравнение, связывающее значение функции u_{ik} и значения $u(x, y)$ в соседних узлах, т. е. получим разностную аппроксимацию дифференциального оператора исходного уравнения.

Аппроксимация граничных условий. Граничные узлы, как правило, не будут находиться на границе Γ . Поэтому необходимо

димо заменить граничные условия для дифференциального уравнения граничными условиями для разностного уравнения, составляемого только для внутренних узлов сетки. Такую аппроксимацию можно осуществлять простым способом, т. е. принять значение функции $u_{i,k}$ в узлах, обозначенных звездочками, равным значению функции $f(x, y)$ в ближайшей точке границы Γ , где $u|_{\Gamma} = f(x, y)$ — краевое условие задачи.

Для более точной аппроксимации граничных условий можно применять линейную интерполяцию, предложенную Коллатцом. При этом для каждого узла A рассмотрим на прямой сетки ближайшую точку B , лежащую на границе Γ , удаленную от A на расстояние $\delta < h$, и ближайший внутренний узел C (рис. 27).

Тогда для узла A полагаем

$$u(A) = \frac{\delta u_c + h f(B)}{\delta + h}.$$

Получаемая при этом погрешность будет порядка h^2 . В случае прямоугольной области D значения функции в граничных точках в точности равны значениям граничной функции.

Оценка погрешности и сходимость метода сеток. Погрешность, получаемая в методе сеток, состоит из погрешности замены дифференциального уравнения разностным уравнением, погрешности аппроксимации граничных условий и погрешности приближенного решения системы разностных уравнений. Разностная задача должна обладать свойствами устойчивости и сходимости. Устойчивость схемы означает, что малым изменениям начальных условий соответствуют малые изменения решения разностной задачи. Сходимость схемы означает, что если неограниченно измельчать сетку, то последовательность решений, получаемых методом сеток, будет сходиться равномерно к точному решению краевой задачи.

9.2. Решение задачи Дирихле для уравнения Лапласа методом сеток

Рассмотрим решение методом сеток задачи Дирихле для уравнения Лапласа — уравнения эллиптического типа. Пусть требуется найти непрерывную функцию $u = u(x, y)$, удовлетворяющую внутри некоторой конечной области D уравнению

$$\Delta u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (9.4)$$

а на границе Γ — условию

$$u(x, y)|_{\Gamma} = f(x, y), \quad (9.5)$$

где $f(x, y)$ — заданная непрерывная на Γ функция. Предполагаем, что Γ — простой замкнутый кусочно-гладкий контур.

Выберем квадратную сетку с шагом h . Воспользуемся формулами (9.3) для составления разностного уравнения, соответствующего внутреннему узлу $(i, k) = (x_i, y_k)$. Множество узлов, значения сеточной функции в которых входят в выражение разностного уравнения в точке (i, k) , называют шаблоном. В данной схеме используется шаблон, когда система узлов расположена по «кресту» (рис. 28). Для каждого внутреннего узла (x_i, y_k) сетки G получим конечно-разностное уравнение

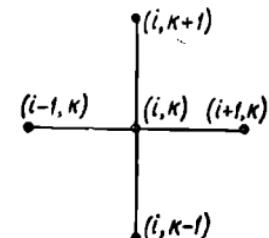


Рис. 28

$$\frac{u_{i-1, k} - 2u_{i, k} + u_{i+1, k}}{h^2} + \frac{u_{i, k+1} - 2u_{i, k} + u_{i, k-1}}{h^2} = 0.$$

Преобразуем его к виду

$$u_{i, k} = \frac{1}{4} (u_{i-1, k} + u_{i+1, k} + u_{i, k+1} + u_{i, k-1}). \quad (9.6)$$

Погрешность замены дифференциального уравнения (9.4) системой разностных уравнений (9.6) имеет порядок $O(h^2)$ и оценивается неравенством

$$|R_{ik}| \leq \frac{h^2}{6} M_1, \text{ где } M_1 = \max_D \left\{ \left| \frac{\partial^4 u}{\partial x^4} \right|; \left| \frac{\partial^4 u}{\partial y^4} \right| \right\}.$$

Описанная здесь разностная схема обладает свойствами устойчивости и сходимости.

Система алгебраических уравнений (9.6) всегда совместна и имеет единственное решение. Его можно получить методом Гаусса. Но если число узлов сетки велико, то удобнее применять итерационные методы (прогонки, простой итерации, Зейделя).

В качестве примера итерационного метода рассмотрим процесс усреднения Либмана для системы (9.6). Выбрав начальные приближения $u_{i,k}^{(0)}$, последовательные приближения $u_{i,k}^{(n)}$ для внутренних узлов сеточной области определяем по формуле

$$u_{i, k}^{(n)} = \frac{1}{4} (u_{i-1, k}^{(n-1)} + u_{i+1, k}^{(n)} + u_{i, k-1}^{(n)} + u_{i, k+1}^{(n-1)}), \quad (9.7)$$

$$n = 1, 2, 3, \dots$$

Итерационный процесс продолжаем до тех пор, пока не совпадут два последовательных приближения $u_{i,k}^{(n+1)}$ и $u_{i,k}^{(n)}$ с заданной точностью.

Известно [15], что для любого шага h процесс Либмана сходится к точному решению независимо от выбора начального приближения. Погрешность приближенного решения имеет порядок $O(h^2)$. Следует иметь в виду, что в силу принципа максимума для значений искомой функции должны быть выполнены неравенства

$$m \leq u_{i,k} \leq M,$$

где $m = \min_D f(x, y)$, $M = \max_D f(x, y)$. Поэтому разумно полагать $m \leq u_{i,k}^{(0)} \leq M$, т. е. выбрать в качестве $u_{i,k}^{(0)}$ среднее арифметическое значений $u(x, y)$ в четырех граничных точках, расположенных на одной горизонтали и вертикали с точкой $u_{i,k}$.

9.3. Решение первой краевой задачи для уравнения параболического типа методом сеток

Рассмотрим численное решение методом сеток уравнения теплопроводности

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}. \quad (9.8)$$

Пусть искомая функция $u(x, t)$ определяет температуру в любой точке x стержня длиной l в момент времени t . В области $D = \{(x, t) | 0 \leq x \leq l, 0 \leq t \leq T\}$ заданы начальные условия (распределение температуры в начальный момент времени)

$$u(x, 0) = f(x), \quad 0 \leq x \leq l \quad (9.9)$$

и краевые условия первого рода (тепловые режимы на концах стержня)

$$\begin{cases} u(0, t) = \varphi(t), \\ u(l, t) = g(t), \\ 0 \leq t \leq T. \end{cases} \quad (9.10)$$

Функции $f(x)$, $\varphi(t)$ и $g(t)$ предполагаются непрерывными. В системе координат xOt построим прямоугольную сетку с шагом h по оси x и с шагом τ по оси ot . Введем обозначения:

$$x_i = ih \quad (i = 0, 1, 2, \dots, n), \quad h = l/n.$$

$$t_k = k\tau \quad (k = 0, 1, 2, \dots, m), \quad \tau = \frac{T}{m}; \quad u(x_i, t_k) = u_{ik}.$$

Заменим частные производные в уравнении (9.8) их разностными отношениями (9.2) и (9.3). При этом используется явная схема выбора узлов, расположенных по шаблону, представленному на рис. 29, а. Уравнение (9.8) аппроксимируется конечно-разностным уравнением

$$\frac{u_{i,k+1} - u_{i,k}}{\tau} = a^2 \frac{u_{i+1,k} - 2u_{i,k} + u_{i-1,k}}{h^2},$$

которое можно привести к виду

$$u_{i,k+1} = \left(1 - \frac{2a^2\tau}{h^2}\right) u_{i,k} + a^2 \frac{\tau}{h^2} (u_{i+1,k} + u_{i-1,k}). \quad (9.11)$$

По формуле (9.11) можно вычислить значение искомой функции $u(x, t)$ в узлах $k+1$ -го временного слоя при $t = t_{k+1}$, если известны ее значения в узлах k -го слоя. При $t = 0$ значения функции получаем из начальных условий (9.9):

$$u_{i,0} = f(x_i) \quad (i = 1, 2, \dots, n-1).$$

В крайних левом и правом узлах каждого слоя ($i = 0; i = n$) значения функции определяются из граничных условий

$$u(0, t_k) = \varphi(t_k), \quad u(l, t_k) = g(t_k), \quad (k = 0, 1, \dots, m).$$

Таким образом, вычисления по явной схеме разностной аппроксимации (9.11) осуществляются поочередно для всех временных слоев достаточно просто, но они обладают одним существенным недостатком. Для устойчивости вычислений по явной схеме (9.11) накладываются дополнительные ограничения на сетку: значения шагов по независимым переменным должны выбираться из соотношения

$$0 < \tau \leq \frac{h^2}{2a^2}. \quad (9.12)$$

В частности, при $\tau = \frac{h^2}{2a^2}$ формула (9.11) существенно упрощается:

$$u_{i,k+1} = \frac{1}{2} (u_{i+1,k} + u_{i-1,k}). \quad (9.13)$$

Ограничение (9.12) на выбор шага по переменной t может сильно увеличить объем вычислений.

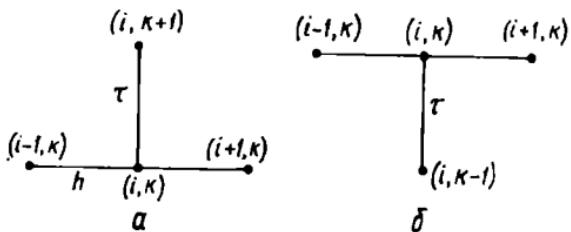


Рис. 29

Выберем в разностной аппроксимации дифференциального оператора другой шаблон (рис. 29,б), применив разностную аппроксимацию

$$\frac{\partial u(x_i, t_k)}{\partial t} \approx \frac{u_{i,k} - u_{i,k-1}}{\tau}. \quad (9.14)$$

Такая схема выбора узлов называется *неявной*.

Уравнение (9.8) заменится системой

$$\frac{u_{i,k} - u_{i,k-1}}{\tau} = a^2 \frac{u_{i+1,k} - 2u_{i,k} + u_{i-1,k}}{h^2},$$

которую можно преобразовать к виду

$$\frac{a^2\tau}{h^2} u_{i+1,k} - \left(1 + 2\frac{a^2\tau}{h^2}\right) u_{i,k} + \frac{a^2\tau}{h^2} u_{i-1,k} = -u_{i,k-1}. \quad (9.15)$$

Из краевых условий (9.10) следуют уравнения

$$u_{0,k} = \varphi(t_k), \quad u_{n,k} = g(t_k), \quad (k = 0, 1, \dots, m). \quad (9.16)$$

Система разностных уравнений (9.15), (9.16) имеет трехдиагональную матрицу и может быть решена методом прогонки. Уравнения (9.15) определяют приближенное значение функции $u(x, t)$ на k -м временном слое ($t = t_k$) по известным ее значениям на $k-1$ -м слое. На нулевом слое ($k = 0$) значения $u_{i,0}$ определяются из начальных условий (9.9): $u_{i,0} = f(x_i)$, ($i = 1, 2, \dots, n-1$).

Применение неявной разностной схемы (9.15) требует решения на каждом шаге системы алгебраических уравнений, но вычисления будут устойчивыми при любом выборе шагов h и τ . Порядок аппроксимации разностной схемы (9.15) равен $O(\tau + h^2)$ [15].

9.4. Решение первой краевой задачи для уравнения гиперболического типа методом сеток

В качестве примера уравнения гиперболического типа рассмотрим уравнение малых поперечных колебаний струны

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, \quad 0 \leq x \leq l, \quad 0 \leq t \leq T. \quad (9.17)$$

В случае первой краевой задачи искомое отклонение струны в любой точке x в момент времени t должно удовлетворять начальным

$$u(x, 0) = \varphi(x), \quad \frac{\partial u(x, 0)}{\partial t} = g(x), \quad 0 \leq x \leq l, \quad (9.18)$$

и краевым условиям

$$u(0, t) = \alpha(t), \quad u(l, t) = \beta(t), \quad 0 \leq t \leq T. \quad (9.19)$$

Построим в области $D = \{(x, y) | 0 \leq x \leq l, 0 \leq t \leq T\}$ прямоугольную сетку $x_i = ih$ ($i = 0, 1, \dots, n$), $t_k = kt$ ($k = 0, 1, \dots, m$) и заменим частные производные в уравнении (9.17) разностными соотношениями на шаблоне типа «крест» (см. рис. 28). Для каждого внутреннего узла сетки получим конечно-разностное уравнение

$$\frac{u_{i,k+1} - 2u_{i,k} + u_{i,k-1}}{\tau^2} = a^2 \frac{u_{i+1,k} - 2u_{i,k} + u_{i-1,k}}{h^2},$$

которое преобразуем к виду

$$u_{i,k+1} = 2 \left(1 - \frac{\tau^2 a^2}{h^2}\right) u_{i,k} + \frac{a^2 \tau^2}{h^2} (u_{i+1,k} + u_{i-1,k}) - u_{i,k-1}. \quad (9.20)$$

Схема вычислений по (9.20) является явной, она позволяет вычислить приближенно значения функции в узлах $(k+1)$ -го слоя по известным ее значениям на двух предыдущих слоях. На первых двух слоях значения функции определяются из начальных условий (9.18). Полагаем

$$u_{i,0} = \varphi(x_i), \quad (i = 0, 1, \dots, n), \quad (9.21)$$

а для производной по времени применяем аппроксимацию (9.2):

$$\frac{\partial u(x_i, 0)}{\partial t} \approx \frac{u_{i1} - u_{i0}}{\tau} = g(x_i), \quad (9.22)$$

отсюда $u_{i1} = u_{i0} + \tau g(x_i)$ ($i = 0, 1, 2, \dots, n$).

Порядок такой разностной аппроксимации равен $O(\tau + h^2)$. Схема вычислений будет устойчивой, если выполняется условие Куранта

$$\tau < h. \quad (9.23)$$

Если $\tau < h$, то решение разностной задачи (9.20)–(9.22) равномерно стремится к решению исходной краевой задачи при $h \rightarrow 0$ и $\tau \rightarrow 0$.

9.5. Лабораторная работа 13. Решение задачи Дирихле для уравнения Лапласа в прямоугольной области методом сеток

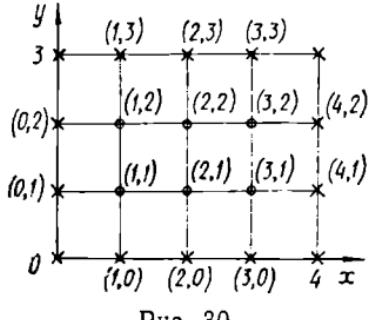
Пусть функция $u(x, y)$ удовлетворяет впограничной прямоугольной области $D = \{(x, y) | 0 \leq x \leq a, 0 \leq y \leq b\}$ уравнению Лапласа (9.4) и принимает на границе Γ следующие значения:

$$\begin{aligned} u(0, y) &= f_1(y), \quad y \in [0, b]; \quad u(a, y) = f_2(y), \quad y \in [0, b]; \\ u(x, 0) &= f_3(x), \quad x \in [0, a]; \quad u(x, b) = f_4(x), \quad x \in [0, a], \end{aligned}$$

где f_i — заданные функции, удовлетворяющие условию непрерывности $u(x, y)$ на границе Γ , т. е.

$$f_1(0) = f_3(0), \quad f_1(b) = f_4(0), \quad f_2(0) = f_3(a), \quad f_2(b) = f_4(a).$$

Программа составлена для случая квадратной сетки с шагом $h = 1$ и числом внутренних узлов, равным 6, т. е. $x_i = x_0 + ih$, ($i = \overline{0, 4}$), $a = 4$, $y_k = y_0 + kh$ ($k = \overline{0, 3}$), $b = 3$ (рис. 30).



Квадратную сетку с шагом h в области $G = \{(x, y) / 0 \leq x \leq c, 0 \leq y \leq d\}$ заменой переменной $\xi = \frac{x}{h}$, $\eta = \frac{y}{h}$ можно превратить в квадратную сетку с шагом 1 в области $D = \{\xi, \eta) / 0 \leq \xi \leq a, 0 \leq \eta \leq b\}$ ($a = \frac{c}{h}$, $b = \frac{d}{h}$).

Инструкция. 1. В режиме программирования (F ПРГ) ввести программу.

2. В автоматическом режиме (F АВТ) ввести граничные условия по схеме (см. рис. 30):

$$\begin{aligned} (u_{02} + u_{13}) \text{ ПА}, \quad (u_{23}) \text{ П7}, \quad (u_{33} + u_{42}) \text{ ПВ}; \\ (u_{30} + u_{41}) \text{ ПС}, \quad (u_{20}) \text{ П8}, \quad (u_{10} + u_{01}) \text{ ПД}. \end{aligned}$$

3. Ввести начальные значения $(u_{ik}^{(0)})$ в регистры 1—6:

$$\left(u_{12}^{(0)} = \frac{1}{4} (u_{02} + u_{13} + u_{42} + u_{10}) \right) \text{ П1},$$

$$\left(u_{22}^{(0)} = \frac{1}{4} (u_{02} + u_{23} + u_{42} + u_{20}) \right) \text{ П2},$$

$$\left(u_{32}^{(0)} = \frac{1}{4} (u_{02} + u_{33} + u_{42} + u_{30}) \right) \text{ П3},$$

$$\left(u_{11}^{(0)} = \frac{1}{4} (u_{01} + u_{13} + u_{41} + u_{10}) \right) \text{ П4},$$

$$\left(u_{21}^{(0)} = \frac{1}{4} (u_{01} + u_{23} + u_{41} + u_{20}) \right) \text{ П5},$$

$$\left(u_{31}^{(0)} = \frac{1}{4} (u_{01} + u_{33} + u_{41} + u_{30}) \right) \text{ П6}.$$

4. Нажать клавиши В/О С/П. После ОСТАНОВа спи-
сать значения $u_{ik}^{(1)}$ в такой последовательности: $u_{31}^{(1)}, u_{21}^{(1)}$,
 $u_{11}^{(1)}, u_{32}^{(1)}, u_{22}^{(1)}, u_{12}^{(1)}$.

Нажать клавишу С/П для вычисления следующего значе-
ния этой последовательности.

5. Повторить п. 4 следующего шага итерации до стабили-
зации значений $u_{ik}^{(n)}$.

Программа 41 решения задачи Дирихле для уравнения
Лапласа (9.4), (9.5) в прямоугольной области методом сеток
по (9.7).

Адрес	Команда	Код	Адрес	Команда	Код	Адрес	Команда	Код
00	7	07	15	ИП1	61	30	38	38
01	ПО	40	16	ИП5	65	31	ИПА	6—
02	ИПС	6С	17	ПП	53	32	ИП2	62
03	ИПЗ	63	18	38	38	33	ИП4	64
04	ИП5	65	19	ИПВ	6L	34	ПП	53
05	ПП	53	20	ИП2	62	35	38	38
06	38	38	21	ИП6	66	36	БП	51
07	ИП8	68	22	ПП	53	37	00	00
08	ИП2,	62	23	38	38	38	+	10
09	ИП4	64	24	ИП1	61	39	+	10
10	ИП6	66	25	ИПЗ	63	40	4	04
11	+	+	26	ИП5	65	41	÷	13
12	ПП	53	27	ИП7	67	42	КПО	10
13	38	38	28	+	10	43	С/П	50
14	ИПД	6Г	29	ПП	53	44	В/О	52

Распределение регистров памяти:

Номер регистра	1	2	3	4	5	6	7	8	A	B	C
Переменная	u_{12}	u_{22}	u_{32}	u_{11}	u_{21}	u_{31}	u_{23}	u_{20}	$u_{02} + u_{13}$	$u_{33} + u_{42}$	$u_{30} + u_{41}$
									D		
									$u_{01} + u_{10}$		

Контрольный пример. Решить задачу Дирихле для уравнения Лап-
ласа (9.4) в области

$$D = \{(x, y) \mid 0 \leq x \leq 4; 0 \leq y \leq 3\}$$

с граничными условиями

$$u|_{x=0} = 0; \quad u|_{y=0} = 0;$$

$$u|_{x=4} = \frac{y^2}{9}, \quad u|_{x=3} = \frac{x^2}{16},$$

выбрав квадратную сетку с шагом $h = 1$.

Вычисляем значения $u(x, y)$ в граничных точках (см. рис. 30);

$$u_{01} = u_{02} = u_{10} = u_{20} = u_{30} = 0; \quad u_{41} = 0,1111; \quad u_{42} = 0,4444,$$

$$u_{13} = 0,0625; \quad u_{23} = 0,2500; \quad u_{33} = 0,5625.$$

Выполняем п. 2. Вычисляем начальные значения и заносим их в память согласно п. 3:

$$u_{12}^{(0)} = 0,12; \quad u_{22}^{(0)} = 0,18; \quad u_{32}^{(0)} = 0,25; \quad u_{11}^{(0)} = 0,04; \quad u_{21}^{(0)} = 0,09;$$

$$u_{31}^{(0)} = 0,17.$$

Нажимаем клавиши В/О С/П, после ОСТАНОВа списываем значение $u_{31}^{(1)} = 0,1128$. Нажимаем клавишу С/П, после ОСТАНОВа списываем значение $u_{21}^{(1)} = 0,0832$ и т. д. (см. п. 4).

Результаты удобно записать в таблицу:

Номер итерации	$u_{31}^{(n)}$	$u_{21}^{(n)}$	$u_{11}^{(n)}$	$u_{32}^{(n)}$	$u_{22}^{(n)}$	$u_{12}^{(n)}$
0	0,17	0,09	0,04	0,25	0,18	0,12
1	0,1128	0,0832	0,0508	0,3249	0,1945	0,0769
2	0,1298	0,0938	0,0427	0,3328	0,1884	0,0734
3	0,1344	0,0914	0,0412	0,3324	0,1868	0,0726
4	0,1337	0,0904	0,0408	0,3319	0,1862	0,0724
5	0,1334	0,0901	0,0406	0,3316	0,1860	0,0723
6	0,1332	0,0900	0,0406	0,3316	0,1859	0,0722
7	0,1331	0,0899	0,0405	0,3315	0,1859	0,0722
8	0,1331	0,0899	0,0405	0,3315	0,1859	0,0722

Для выполнения следующего шага итерации нажимаем клавиши В/О С/П. Вычисления прекращаем, как только в двух последовательных приближениях совпадут три десятичных знака.

Ответ.

$$u_{11} = 0,040 \quad u_{21} = 0,090 \quad u_{31} = 0,133$$

$$u_{12} = 0,072 \quad u_{22} = 0,186 \quad u_{32} = 0,332$$

Задания к лабораторной работе 13. Решить задачу Дирихле для уравнения Лапласа (9.4) в области $0 < x < 4$, $0 < y < 3$ с данными граничными условиями, рассмотрев квадратную сетку с шагом $h = 1$. Итерационный процесс продолжить до совпадения трех десятичных знаков.

Номер варианта	$u _{x=0}$	$u _{y=0}$	$u _{x=4}$	$u _{y=3}$
1	$-y^2$	x^2	$16 - 8y - y^2$	$x^2 - 6x - 9$
2	$-y^2$	x^2	$16 + 8y - y^2$	$x^2 + 6x - 9$
3	$-y^3$	0	$48y - y^3$	$9x^2 - 27$
4	0	0	$4y$	$3x$
5	$4y - 12$	$3x - 12$	$4y$	$3x$
6	$12 - 4y$	$12 - 3x$	0	0
7	$-4y$	0	0	$3x - 12$
8	$-4y$	$3x$	$12 - 4y$	$3x - 12$
9	0	x^3	$64 - 12y^2$	$x^3 - 27x$
10	$-y^3$	0	$52y - y^3$	$9x^2 + 3x - 27$
11	y^2	x^2	$16 + y^2 - 4y$	$x^2 + 9 - 3x$
12	y^2	$x^2 - 3x$	$14 + y^2$	$x^2 + 9 - 3x$
13	$12 - 4y$	$12 - 3x$	$4y$	$3x$
14	0	0	$16y - 4y^2$	$3x^2 - 9x$
15	0	$3x$	$12 - 4y$	0
16	y^2	$-x^2$	$y^2 + 8y - 16$	$9 + 6x - x^2$
17	y^2	$-x^2$	$y^2 - 8y - 16$	$9 - 6x - x^2$
18	y^3	0	$y^3 - 48y$	$27 - 9x^2$
19	0	0	$-4y$	$-3x$
20	$12 - 4y$	$12 - 3x$	$-4y$	$-3x$
21	$4y - 12$	$3x - 12$	0	0
22	$4y$	0	0	$12 - 3x$
23	$4y$	$-3x$	$4y - 12$	$12 - 3x$
24	0	$-x^3$	$12y^2 - 64$	$27x - x^3$
25	y^3	0	$y^3 - 52y$	$27 - 9x^2 - 3x$
26	$-y^2$	$-x^2$	$4y - 16 - y^2$	$3x - x^2 - 9$
27	$-y^2$	$3x - x^2$	$-14 - y^2$	$3x - x^2 - 9$
28	$4y - 12$	$3x - 12$	$-4y$	$-3x$
29	0	0	$4y^2 - 16y$	$9x - 3x^2$
30	0	$-3x$	$4y - 12$	0

Глава 10. ВЫЧИСЛЕНИЕ ЧИСЛОВЫХ ХАРАКТЕРИСТИК СЛУЧАЙНЫХ ВЕКТОРОВ

10.1. Аксиоматическое определение вероятности

Пусть в результате некоторого опыта (испытания) наступит одно из событий ω_i . Совокупность всех таких событий Ω называют основным пространством (пространством элементарных событий), а сами события ω_i — элементарными событиями. Все дальнейшие построения осуществляют для случая конечного (содержащего конечное количество элементов) множества Ω :

$$\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}.$$

Случайным событием называется подмножество A основного пространства Ω . Пустое множество \emptyset и само множество

Ω , рассматриваемые как подмножества основного пространства, называются соответственно невозможным и достоверным событиями.

Определим на множестве $\mathcal{A} = \{A : A \subset \Omega\}$ событий (подмножеств основного пространства Ω) функцию $p(A)$, удовлетворяющую аксиоматике:

- 1) $p(A) > 0, A \in \mathcal{A};$
- 2) $p(\Omega) = 1;$
- 3) если $A \cap B = \emptyset (A, B \in \mathcal{A})$, то $p(A \cup B) = p(A) + p(B).$

Определенная таким образом функция p называется вероятностью (вероятностной мерой), а ее значение $p(A)$ — вероятностью события A .

Из аксиоматики вероятности вытекают следующие ее свойства:

$$p(A) \leq 1; p(\emptyset) = 0; p(A \cup B) = p(A) + p(B) - p(A \cap B);$$

если $A = \bigcup_{i=1}^m \omega_{ki}$, то $p(A) = \sum_{i=1}^m p(\omega_{ki})$. Последнее свойство означает, что достаточно задать вероятность на элементарных событиях.

Совокупность (Ω, \mathcal{A}, p) , где Ω — множество элементарных событий, \mathcal{A} — совокупность всех подмножеств множества Ω , $p = p(\mathcal{A})$ — вероятность, определенная на множестве \mathcal{A} и удовлетворяющая аксиомам 1—3, называется вероятностным пространством. Отметим, что для бесконечного множества Ω определение вероятности несколько усложняется.

Результативность применения аксиоматического определения связана с тем, что для многих реальных явлений характерна устойчивость частот событий при повторении опытов. Конкретные значения вероятностей на элементарных событиях могут вычисляться либо приближенно, как частота при достаточно большом количестве опытов (статистическая вероятность), либо из соображений симметрии элементарных событий (классическая вероятность), либо из иных теоретических соображений.

10.2. Случайные величины и случайные векторы

Пусть (Ω, \mathcal{A}, p) — вероятностное пространство.

Случайной величиной X называется функция $f = f(\omega)$, определенная на множестве Ω элементарных событий. Значе-

ния этой функции $x_i = f(\omega_i)$ называются значениями случайной величины X . Если одно значение случайной величины x_i отвечает совокупности элементарных событий $\omega_{i1}, \dots, \omega_{ik}$, то можно построить новое вероятностное пространство с элементарными событиями

$$\tau_i = \{\omega_{i1}, \dots, \omega_{ik}\}; p(\tau_i) = \sum_{q=1}^k p(\omega_{iq}).$$

При изучении свойств случайной величины можно абстрагироваться от конкретных случайных событий и рассматривать только значения случайной величины $x_i = f(\omega_i)$ и отвечающие им вероятности $p_i = p(\omega_i)$. Таким образом, достаточно задать совокупность пар чисел (x_i, p_i) ($i = 1, \dots, n$; $p\{X = x_i\} = p_i$), где $p_i \geq 0$, $\sum_{i=1}^n p_i = 1$. Такая совокупность называется законом распределения случайной величины X .

Совокупность l случайных величин называется l -мерным случальным вектором. Закон распределения, например, двумерного вектора (X, Y) задается совокупностью троек чисел (x_i, y_i, p_{ij}) ($i = 1, \dots, n$; $j = 1, \dots, m$), где $p_{ij} \geq 0$; $\sum_{i,j} p_{ij} = 1$.

Здесь p_{ij} — вероятность того, что случайная величина X примет значение x_i , а Y — y_j . По закону распределения случайного вектора (X, Y) можно восстановить закон распределения (x_i, p_i) случайной величины X : $p_i = \sum_{j=1}^m p_{ij}$, либо аналогично закон распределения случайной величины Y .

10.3. Числовые характеристики случайного двухмерного вектора

Свойства случайной величины X описываются моментами k -го порядка ($k = 1, 2, \dots$):

$$M^k X = \sum_{i=1}^n p_i x_i^k, \quad k = 1, 2, \dots; \tag{10.1}$$

$$\dot{M}^k X = \sum_{i=1}^n p_i (x_i - MX)^k, \quad k = 2, 3, \dots,$$

где MX — момент 1-го порядка.

Момент \dot{M}^k ($k = 2, 3, \dots$) называется центральным моментом k -го порядка, а случайная величина $X - MX$ — центрированной случайной величиной.

Момент первого порядка MX называется *математическим ожиданием* случайной величины X :

$$MX = \sum_{i=1}^n p_i x_i. \quad (10.2)$$

Центральный момент второго порядка случайной величины X называется *дисперсией*

$$DX = \sum_{i=1}^n p_i (x_i - MX)^2 = \sum_{i=1}^n p_i x_i^2 - (MX)^2, \quad (10.3)$$

а корень квадратный из дисперсии — *средним квадратическим отклонением*:

$$\sigma_X = \sqrt{DX}. \quad (10.4)$$

Случайная величина $X^* = \frac{X}{\sigma_X}$ называется *нормированной*.

Ковариацией случайных величин X и Y называется математическое ожидание произведения центрированных случайных величин:

$$K(X, Y) = M[(X - MX)(Y - MY)] = \sum_{i,j} p_{ij} (x_i - MX) \times \\ \times (y_j - MY) = \sum_{i,j} p_{ij} x_i y_j - MX MY. \quad (10.5)$$

Ковариация нормированных случайных величин X^* и Y^* называется *коэффициентом корреляции* случайных величин X и Y :

$$\rho(X, Y) = \frac{K(X, Y)}{\sigma_X \sigma_Y}. \quad (10.6)$$

10.4. Лабораторная работа 14. Вычисление числовых характеристик двухмерного случайного вектора

Вычисление математических ожиданий MX и MY , дисперсий DX и DY , средних квадратических отклонений σ_X , σ_Y , ковариации $K(X, Y)$ и коэффициента корреляции $\rho(X, Y)$ (10.2)–(10.6) осуществляется согласно программе 42.

Программа 42 вычисления числовых характеристик двухмерного случайного вектора по (10.2)–(10.6).

Адрес	Команда	Код	Содержание операции
00	ИП1	61	
01	П2	42	
02	0	00	
03	ПА	4—	
04	ПВ	4Л	
05	С/П	50	ОСТАНОВ для ввода x_i
06	П8	48	ОСТАНОВ для ввода y_j
07	С/П	50	ОСТАНОВ для ввода p_{ij}
08	П9	49	
09	С/П	50	
10	ПД	4Г	
11	ИПА	6—	
12	+	10	Вычисление $\sum_{j=1}^m p_{ij}$
13	ПА	4—	
14	ИП9	69	
15	ИПД	6Г	
16	×	12	
17	ПД	4Г	Вычисление $\sum_{j=1}^m y_j p_{ij}$
18	ИПВ	6Л	
19	+	10	
20	ПВ	4Л	
21	ИП9	69	
22	ИПД	6Г	
23	×	12	
24	ПД	4Г	Вычисление $M(Y^2)$
25	ИПЗ	63	
26	+	10	
27	П3	43	
28	FL2	58	
29	07	07	Цикл по индексу j
30	ИПВ	6Л	
31	ИП5	65	Вычисление MY
32	+	10	
33	П5	45	
34	ИП8	68	
35	ИПА	6—	
36	×	12	
37	ПД	4Г	Вычисление MX
38	ИП7	67	
39	+	10	
40	П7	47	
41	ИП8	68	
42	ИПД	6Г	
43	×	12	
44	ИП6	66	Вычисление $M(X^2)$
45	+	10	
46	П6	46	

Адрес	Команда	Код	Содержание операции
47	ИП8	68	
48	ИПВ	6Л	
49	Х	12	
50	ИП4	64	Вычисление $M(X, Y)$
51	+	10	
52	П4	44	
53	FLO	5Г	
54	00	00	Цикл по индексу i
55	ИП7	67	
56	ИП5	65	Вычисление $K(X, Y)$
57	Х	12	
58	—	11	
59	ПС	4С	
60	ИП3	63	
61	ИП5	65	
62	FX ²	22	Вычисление DY
63	—	11	
64	ПВ	4Л	
65	ИП6	66	
66	ИП7	67	
67	FX ²	22	Вычисление DX
68	—	11	
69	ПА	4—	
70	Х	12	
71	FV	21	
72	÷	13	
73	ПД	4Г	Вычисление $\rho(X, Y)$
74	С/П	50	

Инструкция. 1. Нажав клавиши F ПРГ, ввести программу.

2. В автоматическом режиме (F АВТ) очистить регистры памяти:

0 П3 П4 П5 П6 П7

и ввести значения n и m

(n) ПО (m) П1

3. Нажать клавиши В/О С/П и вводить данные по схеме:

(x_1) С/П (y_1) С/П (p_{11}) С/П (y_2) С/П (p_{12}) С/П ...

(y_m) С/П (p_{1m}) С/П

(x_2) С/П (y_1) С/П (p_{21}) С/П (y_2) С/П (p_{22}) С/П ...

(y_m) С/П (p_{2m}) С/П

(x_n) С/П (y_1) С/П (p_{n1}) С/П (y_2) С/П (p_{n2}) С/П

(y_m) С/П (p_{nm}) С/П

(ввод каждого числа проводить после ОСТАНОВА).

Результаты вычислений хранятся в регистрах:

Номер регистра	7	6	5	3	1	A	B	C	D
Перемен- ная	MX	$M(X^2)$	$M(Y)$	$M(Y^2)$	$M(X, Y)$	$D(X)$	$D(Y)$	$K(X, Y)$	$\rho(X, Y)$

Контрольный пример. Двухмерный случайный вектор (X, Y) задан таблицей вероятностей p_{ij} :

x_i	y_j	
	1	2
-1	0,1	0,2
0	0,2	0,2
1	0,2	0,1

Найти числовые характеристики случайного вектора.

1. Нажимаем клавиши F ПРГ и клавиши программы.
2. В автоматическом режиме (F АВТ) вводим данные и приступаем к вычислениям:

0	П3	ПИ	П5	П6	П7	3	П0	2	П1	В/О	С/П
1	/—/	С/П	1	С/П	0,1	С/П	2	С/П	0,2	С/П	
	0	С/П	1	С/П	0,2	С/П	2	С/П	0,2	С/П	
	1	С/П	1	С/П	0,2	С/П	2	С/П	0,1	С/П	

Списываем с индикатора значение $\rho(X, Y) = -0,2582$.

Из соответствующих регистров вызываем значения числовых характеристик: ИП7 ($MX = 0$) ИП5 ($MY = 1,5$) ИПА ($DX = 0,6$) ИПВ ($DY = 0,25$) ИПС ($K(X, Y) = -0,1$)

Задания к лабораторной работе 14.

Координаты случайного вектора (X, Y) равны соответственно: $x_i = 0,1n + 0,1$ ($i = 1$), $y_j = 0,1j$ (n — номер варианта; $i = 1, \dots, 4$; $j = 1, \dots, 4$), вероятности заданы таблицей

$i \backslash j$	1	2	3	4
1	0,00	0,00	0,05	0,04
2	0,05	0,06	0,06	0,07
3	0,10	0,19	0,14	0,05
4	0,05	0,05	0,05	0,04

для вариантов 1—15 и таблицей

$i \backslash j$	1	2	3	4
1	0,04	0,05	0,05	0,05
2	0,05	0,14	0,19	0,10
3	0,07	0,06	0,06	0,05
4	0,04	0,05	0,00	0,00

для вариантов 16—30.

Найти математические ожидания MX , MY , дисперсии DX , DY , корреляционный момент $K(X, Y)$ и коэффициент корреляции $\rho(X, Y)$.

Ответы к заданиям по лабораторным работам

Лабораторная работа 1

Номер варианта	x_1	x_2	x_3	Номер варианта	x_1	x_2	x_3
1	0,28	0,11	0,09	16	1,61	-1,55	-2,20
2	-0,12	0,19	0,39	17	1,25	-0,28	-0,31
3	-1,23	0,55	1,30	18	0,98	-0,88	-0,66
4	0,86	0,88	-0,38	19	2,18	0,66	-2,13
5	-0,62	0,66	0,52	20	15,66	-17,24	-4,96
6	-2,48	3,25	2,15	21	0,95	-0,09	0,22
7	1,93	-0,29	-0,42	22	0,59	0,18	-0,61
8	-0,51	0,71	0,26	23	0,39	-0,36	-0,56
9	0,32	0,01	-0,23	24	-0,84	0,98	-1,57
10	0,29	0,22	0,18	25	1,14	1,11	-0,01
11	1,88	2,68	2,65	26	0,31	0,30	0,04
12	0,63	0,47	-0,25	27	0,87	1,02	0,99
13	0,61	-0,08	-0,28	28	0,60	0,50	1,18
14	-0,01	-0,13	0,44	29	0,94	0,75	-0,31
15	-34,98	2,88	54,04	30	0,83	0,54	1,02

Лабораторная работа 2

Номер варианта	x	Номер варианта	x
1	-0,567	16	-2,532
2	1,310	17	0,824
3	1,114	18	-1,532
4	-0,696	19	0,998
5	1,259	20	-1,478
6	1,213	21	-0,121
7	1,174	22	1,059
8	0,567	23	0,424
9	0,696	24	-0,532
10	-1,259	25	-1,864
11	-1,174	26	-0,357
12	0,567	27	0,683
13	-1,213	28	-0,682
14	0,399	29	-1,696
15	-0,824	30	1,695

Лабораторная работа 3

Номер варианта	Решение системы	Количество итераций	Номер варианта	Решение системы	Количество итераций
1	-0,9284; 1,3886	8	16	-0,1381; -0,1381	11
2	0,4512; 0,4915	7	17	0,0854; -0,0854	8
3	0,9639; 0,8833	13	18	-0,8971; -1	4
4	0,1322; 0,1359	10	19	-0,9284; -1,3886	8
5	0,0936; 0,1230	7	20	0,4512; -0,4915	7
6	-0,0975; 0,0837	6	21	0,9639; -0,8833	13
7	0,1381; -0,1381	11	22	0,1322; -0,1359	10
8	0,9284; 1,3886	8	23	0,0936; -0,1230	7
9	-0,0854; -0,0854	4	24	-0,0975; -0,0837	6
10	0,8971; 1	8	25	0,1381; 0,1381	11
11	-0,4512; 0,4915	7	26	-0,0854; 0,0854	8
12	-0,9639; 0,8833	13	27	0,8971; -1	4
13	-0,1322; 0,1359	10	28	0,9994; -0,0012	3
14	-0,0936; 0,1230	7	29	3,2926; -0,1436	8
15	0,0975; 0,0837	6	30	-0,0936; -0,1230	7

Лабораторная работа 4

Номер варианта	Решение системы	Номер варианта	Решение системы
1	1,1961; 0,8360	14	0,3106; -1,0100
2	0,5960; 1,6778	15	-0,4427; -1,0281
3	-0,8723; 1,1908	16	0,1579; 0,9999
4	0,4263; 1,5316	17	-1,1961; 0,8360
5	0,4800; 0,9617	18	-0,5960; 1,6778
6	-1,0100; 0,3106	19	0,8723; 1,1908
7	-1,0281; -0,4427	20	-0,4263; 1,5316
8	0,9999; 0,1579	21	-0,4800; 0,9617
9	0,8360; 1,1961	22	1,0100; 0,3106
10	1,6778; 0,5960	23	1,0281; -0,4427
11	1,1908; -0,8723	24	-0,9999; 0,1579
12	1,5316; 0,4263	25	1,1961; -0,8360
13	0,9617; 0,4800	26	0,4263; -1,5316

Лабораторная работа 5

Номер варианта	Полином Лагранжа		Полином Ньютона	
	x	$P(x)$	x	$P(x)$
1	0,967	2,6318	0,033	1,0316
2	0,934	2,5465	0,066	1,0650
3	0,901	2,4641	0,099	1,0999
4	0,868	2,3844	0,132	1,1365
5	0,835	2,3073	0,165	1,1746
6	0,802	2,2327	0,198	1,2143
7	0,769	2,1605	0,231	1,2556
8	0,736	2,0906	0,264	1,2983
9	0,703	2,0229	0,297	1,3427
10	0,670	1,9573	0,330	1,3885
11	0,637	1,8937	0,363	1,4360
12	0,604	1,8321	0,396	1,4850
13	0,571	1,7724	0,429	1,5356
14	0,538	1,7145	0,462	1,5878
15	0,505	1,6584	0,495	1,6417
16	0,472	1,6039	0,528	1,6973
17	0,439	1,5512	0,561	1,7546
18	0,406	1,5001	0,594	1,8138
19	0,373	1,4506	0,627	1,8748
20	0,340	1,4027	0,660	1,9378
21	0,307	1,3564	0,693	2,0028
22	0,274	1,3116	0,726	2,0699
23	0,241	1,2684	0,759	2,1391
24	0,208	1,2266	0,792	2,2106
25	0,175	1,1865	0,825	2,2845
26	0,142	1,1479	0,854	2,3514
27	0,109	1,1108	0,891	2,4397
28	0,076	1,0754	0,924	2,5212
29	0,043	1,0416	0,957	2,6056
30	0,010	1,0094	0,990	2,6929

Лабораторная работа 6

1. Коэффициенты линейной регрессии $\varphi(x) = c_0 + c_1x$ равны: $c_1 = 1,21$ для всех вариантов, $c_0 = 0,04 + 0,1(k - 1)$, где $k = 1, 2, \dots, 30$ — номер варианта.

2. Коэффициенты квадратичного аппроксимирующего многочлена $y = c_2x^2 + c_1x + c_0$ равны: $c_2 = -0,01$; $c_1 = 0,40$ для всех вариантов; $c_0 = k - 1$ для $k = 1, 2, \dots, 10$; $c_0 = 0,1l - 1$ для $l = 11, 12, \dots, 19$; $c_0 = 0,2r - 2,9$ для $r = 20, 21, \dots, 30$.

Лабораторная работа 7

1. Значения коэффициентов аппроксимирующего тригонометрического полинома приведены в таблице:

Номер варианта	a_0	a_1	b_1	a_2	b_2	a_3	b_3
1	2,5321320	1,1077879	-0,2245564	0,2500798	-0,1056834	0,0368864	-0,0245783
2	2,5705927	1,0845726	-0,4585502	0,2030523	-0,2090754	0,0175342	-0,04520395
3	2,610645	1,0116406	-0,6921065	0,1131374	-0,2911903	-0,0130757	-0,0527429
4	2,6523212	0,8882906	-0,9146170	-0,0097512	-0,3341164	-0,0446585	-0,0410484
5	2,6956552	0,7160684	-1,1151959	-0,1484272	-0,3245472	-0,0647336	-0,0060457
6	2,7406847	0,4988690	-1,2831708	-0,2805481	-0,2569586	-0,0634128	0,031523
7	2,7874515	0,2429330	-1,4085625	-0,3818514	-0,1356176	-0,0375550	0,0695120
8	2,835992	-0,0433052	-1,4825621	-0,4302561	0,0251215	0,0072849	0,0871245
9	2,886343	-0,3494801	-1,4980356	-0,4103068	0,2022685	0,0577018	0,0730156
10	2,9385522	-0,66336092	-1,4499983	-0,3169827	0,3671075	0,0958407	0,02726805
11	2,9926705	-0,9724675	-1,3359588	-0,1577769	0,4893310	0,1047634	-0,029166
12	3,0487362	-1,2621118	-1,256126	-0,0367068	0,47521	0,0542206	-0,0748834
13	3,1067877	-1,5185367	-0,913555838	0,2690952	0,5081114	0,0097195	-0,12757986
14	3,1668872	-1,7283012	-0,6144453	0,47088145	0,3830630	-0,0718590	-0,11563254
15	3,2290987	-1,8790437	-0,2678071	0,6156786	0,1787262	-0,1397050	-0,0593501

Продолжение табл.

Номер награнта	a_n	a_1	b_1	α_z	b_z	α_s	b_s
16	3,29345557	-1,9600373	0,1145857	0,6715676	-0,0785651	-0,1632319	0,0269125
17	3,35999965	-1,962908	0,5187699	0,6187970	-0,3510671	-0,1253517	0,1168367
18	3,428808	-1,8822876	0,9288895	0,4550976	-0,5943076	-0,0326224	0,1787209
19	4,4999707	-1,7160915	1,3275893	0,1974256	-0,7632346	0,0846162	0,1845776
20	3,573501	-1,4655881	1,6968021	-0,1201108	-0,8201500	0,1832803	0,1209582
21	3,6494312	-1,135673	2,0187776	-0,4506608	-0,7432735	0,2245644	-0,0014077
22	3,7279065	-0,7352398	2,27681	-0,7401784	-0,532900	0,1886780	-0,1448152
23	3,8090082	-0,2770262	2,4556082	-0,9346656	-0,2126519	0,0814020	-0,2540406
24	3,892723	0,2231592	2,5419725	-0,9899808	0,1742989	-0,6796102	-0,2778970
25	3,9790875	0,7470729	2,5260352	-0,882515	0,5707181	-0,2130041	-0,1942192
26	4,068301	1,2741177	2,4021812	-0,6162724	0,9116917	-0,3055842	-0,0245402
27	4,160479	1,7816623	2,168947	-0,2229625	1,1330028	-0,2996209	0,1714987
28	4,25553	2,2464433	1,8287652	0,2435993	1,1833195	-0,1785892	0,3203779
29	4,3534985	2,6462915	1,388679	0,71450337	1,0375467	0,0334850	0,3646956
30	4,4547002	2,9607065	0,8612696	0,8612696	1,1125084	0,704778	0,26781035

2. Значения аппроксимирующего тригонометрического полинома $\varphi_3(x)$ в точке $x = \frac{n}{5}$ равны:

Номер варианта n	$\varphi_3\left(\frac{n}{5}\right)$	Номер варианта n	$\varphi_3\left(\frac{n}{5}\right)$
1	0,4533297	16	0,9499335
2	0,5449081	17	0,9610482
3	0,61175846	18	0,9898851
4	0,65219073	19	1,0281999
5	0,6743552	20	1,0591598
6	0,6916536	21	1,062264
7	0,7168135	22	1,0201587
8	0,7568104	23	0,9250952
9	0,8104163	24	0,7827991
10	0,8691371	25	0,6123101
11	0,9210287	26	0,4416598
12	0,9558270	27	0,3006266
13	0,9693574	28	0,2127956
14	0,9654699	29	0,1893959
15	0,9546787	30	0,2268240

Лабораторная работа 8

1. Коэффициенты c_0 , c_1 , c_2 аппроксимации функции $f(x)$ полиномами Лежандра до второго порядка приведены в таблице:

Номер варианта	c_0	c_1	c_2	Номер варианта	c_0	c_1	c_2
1	1,23	0,00	0,65	16	0,29	-0,08	-0,10
2	1,04	0,00	-0,12	17	2,47	1,50	1,30
3	0,96	-0,33	0,13	18	1,23	1,50	0,05
4	0,68	0,00	-0,48	19	0,95	0,42	-0,24
5	0,00	0,59	0,00	20	0,73	-0,20	0,06
6	0,46	-0,21	-0,10	21	1,09	0,00	0,33
7	0,00	1,49	0,00	22	0,55	-0,21	0,16
8	1,23	-1,50	0,65	23	2,06	2,06	0,08
9	0,30	0,14	-0,12	24	0,72	-0,63	-0,01
10	1,60	-2,06	1,90	25	0,64	-0,30	-0,24
11	0,55	0,00	0,11	26	0,72	0,63	-0,01
12	0,61	0,00	-0,19	27	0,88	0,00	-0,20
13	0,96	0,33	0,13	28	0,00	0,65	0,00
14	0,37	0,00	-0,22	29	1,70	0,60	-0,08
15	0,00	0,35	0,00	30	0,62	-0,24	0,11

2.1. Значения коэффициентов Харда до третьего порядка для функции $f(x)$ приведены в таблице:

Номер варианта	C_{01}	C_{11}	C_{21}	C_{22}	C_{31}	C_{32}	C_{33}	C_{44}
1	$8,93 \cdot 10^{-1}$	$8,17 \cdot 10^{-1}$	$5,04 \cdot 10^{-1}$	$4,89 \cdot 10^{-2}$	$1,22 \cdot 10^{-1}$	$1,79 \cdot 10^{-1}$	$4,16 \cdot 10^{-2}$	$2,72 \cdot 10^{-3}$
2	$8,18 \cdot 10^{-1}$	$8,76 \cdot 10^{-1}$	$4,72 \cdot 10^{-1}$	$6,65 \cdot 10^{-2}$	$8,08 \cdot 10^{-2}$	$1,94 \cdot 10^{-1}$	$5,47 \cdot 10^{-2}$	$4,19 \cdot 10^{-3}$
3	$1,07$	$9,23 \cdot 10^{-1}$	$4,19 \cdot 10^{-1}$	$8,88 \cdot 10^{-2}$	$3,55 \cdot 10^{-2}$	$2,05 \cdot 10^{-1}$	$7,01 \cdot 10^{-2}$	$6,33 \cdot 10^{-3}$
4	$1,15$	$9,55 \cdot 10^{-1}$	$3,49 \cdot 10^{-1}$	$1,16 \cdot 10^{-1}$	$-1,19 \cdot 10^{-2}$	$2,09 \cdot 10^{-1}$	$8,79 \cdot 10^{-2}$	$9,35 \cdot 10^{-3}$
5	$1,22$	$9,69 \cdot 10^{-1}$	$2,62 \cdot 10^{-1}$	$1,49 \cdot 10^{-1}$	$-5,86 \cdot 10^{-2}$	$2,05 \cdot 10^{-1}$	$1,08 \cdot 10^{-1}$	$1,36 \cdot 10^{-2}$
6	$1,29$	$9,62 \cdot 10^{-1}$	$1,62 \cdot 10^{-1}$	$1,88 \cdot 10^{-1}$	$-1,02 \cdot 10^{-1}$	$1,92 \cdot 10^{-1}$	$1,28 \cdot 10^{-1}$	$1,92 \cdot 10^{-2}$
7	$1,35$	$9,34 \cdot 10^{-1}$	$5,50 \cdot 10^{-2}$	$2,30 \cdot 10^{-1}$	$-1,40 \cdot 10^{-1}$	$1,70 \cdot 10^{-1}$	$1,50 \cdot 10^{-1}$	$2,66 \cdot 10^{-2}$
8	$1,40$	$8,83 \cdot 10^{-1}$	$-5,50 \cdot 10^{-2}$	$2,76 \cdot 10^{-1}$	$-1,70 \cdot 10^{-1}$	$1,40 \cdot 10^{-1}$	$1,70 \cdot 10^{-1}$	$3,61 \cdot 10^{-2}$
9	$1,44$	$8,09 \cdot 10^{-1}$	$-1,62 \cdot 10^{-1}$	$3,25 \cdot 10^{-1}$	$-1,92 \cdot 10^{-1}$	$1,02 \cdot 10^{-1}$	$1,87 \cdot 10^{-1}$	$4,78 \cdot 10^{-2}$
10	$1,48$	$7,13 \cdot 10^{-1}$	$-2,62 \cdot 10^{-1}$	$3,75 \cdot 10^{-1}$	$-2,05 \cdot 10^{-1}$	$5,86 \cdot 10^{-2}$	$2,00 \cdot 10^{-1}$	$6,21 \cdot 10^{-2}$
11	$1,50$	$5,98 \cdot 10^{-1}$	$-3,49 \cdot 10^{-1}$	$4,21 \cdot 10^{-1}$	$-2,09 \cdot 10^{-1}$	$1,19 \cdot 10^{-2}$	$2,08 \cdot 10^{-1}$	$7,87 \cdot 10^{-2}$
12	$1,53$	$4,65 \cdot 10^{-1}$	$-4,19 \cdot 10^{-1}$	$4,62 \cdot 10^{-1}$	$-2,05 \cdot 10^{-1}$	$-3,55 \cdot 10^{-2}$	$2,08 \cdot 10^{-1}$	$9,75 \cdot 10^{-2}$
13	$1,54$	$3,18 \cdot 10^{-1}$	$-4,71 \cdot 10^{-1}$	$4,94 \cdot 10^{-1}$	$-1,94 \cdot 10^{-1}$	$-8,08 \cdot 10^{-2}$	$1,99 \cdot 10^{-1}$	$1,18 \cdot 10^{-1}$
14	$1,55$	$1,61 \cdot 10^{-1}$	$-5,05 \cdot 10^{-1}$	$5,14 \cdot 10^{-1}$	$-1,79 \cdot 10^{-1}$	$-1,22 \cdot 10^{-1}$	$1,82 \cdot 10^{-1}$	$1,39 \cdot 10^{-1}$
15	$1,55$	0	$-5,18 \cdot 10^{-1}$	$5,18 \cdot 10^{-1}$	$-1,60 \cdot 10^{-1}$	$-1,56 \cdot 10^{-1}$	$1,56 \cdot 10^{-1}$	$1,60 \cdot 10^{-1}$

16	1,55	-1,61·10 ⁻¹	-5,14·10 ⁻¹	5,05·10 ⁻¹	-1,39·10 ⁻¹	-1,82·10 ⁻¹	1,22·10 ⁻¹
17	1,54	-3,18·10 ⁻¹	-4,94·10 ⁻¹	4,72·10 ⁻¹	-1,18·10 ⁻¹	-1,99·10 ⁻¹	8,08·10 ⁻²
18	1,53	-4,65·10 ⁻¹	-4,62·10 ⁻¹	4,19·10 ⁻¹	-9,15·10 ⁻²	-2,08·10 ⁻¹	3,55·10 ⁻²
19	1,50	-5,98·10 ⁻¹	-4,21·10 ⁻¹	3,49·10 ⁻¹	-7,87·10 ⁻²	-2,08·10 ⁻¹	-1,19·10 ⁻²
20	1,48	-7,13·10 ⁻¹	-3,75·10 ⁻¹	2,62·10 ⁻¹	-6,21·10 ⁻²	-2,00·10 ⁻¹	-5,86·10 ⁻²
21	1,44	-8,09·10 ⁻¹	-3,25·10 ⁻¹	1,62·10 ⁻¹	-4,78·10 ⁻²	-1,87·10 ⁻¹	-1,02·10 ⁻¹
22	1,40	-8,83·10 ⁻¹	-2,76·10 ⁻¹	5,50·10 ⁻²	-3,61·10 ⁻²	-1,70·10 ⁻¹	-1,40·10 ⁻¹
23	1,35	-9,34·10 ⁻¹	-2,30·10 ⁻¹	-5,50·10 ⁻²	-2,66·10 ⁻²	-1,50·10 ⁻¹	-1,70·10 ⁻¹
24	1,29	-9,62·10 ⁻¹	-1,87·10 ⁻¹	-1,62·10 ⁻¹	-1,92·10 ⁻²	-1,28·10 ⁻¹	-1,92·10 ⁻¹
25	1,22	-9,69·10 ⁻¹	-1,49·10 ⁻¹	-2,62·10 ⁻¹	-1,35·10 ⁻²	-1,08·10 ⁻¹	-2,05·10 ⁻¹
26	1,15	-9,55·10 ⁻¹	-1,16·10 ⁻¹	-3,49·10 ⁻¹	-9,35·10 ⁻³	-8,79·10 ⁻²	-2,08·10 ⁻¹
27	1,07	-9,23·10 ⁻¹	-8,88·10 ⁻²	-4,19·10 ⁻¹	-6,33·10 ⁻³	-7,01·10 ⁻²	-2,05·10 ⁻¹
28	9,82·10 ⁻¹	-8,76·10 ⁻¹	-6,65·10 ⁻²	-4,72·10 ⁻¹	-4,19·10 ⁻³	-5,46·10 ⁻²	-1,94·10 ⁻¹
29	8,93·10 ⁻¹	-8,17·10 ⁻¹	-4,89·10 ⁻²	-5,04·10 ⁻¹	-2,12·10 ⁻³	-4,16·10 ⁻²	-1,19·10 ⁻¹
30	8,03·10 ⁻¹	-7,48·10 ⁻¹	-3,52·10 ⁻²	-5,18·10 ⁻¹	-1,73·10 ⁻³	-3,10·10 ⁻²	-1,60·10 ⁻¹

2Б. Значения аппроксимирующей ступенчатой функции Хаара третьего порядка $g_3(x)$ на отрезках $\left[\frac{i-1}{2^3}; \frac{i}{2^3}\right]$ ($i = 1, 2, \dots, 8$) приведены в таблице:

Номер варианта	$\left[0; \frac{1}{8}\right]$	$\left[\frac{1}{8}; \frac{2}{8}\right]$	$\left[\frac{2}{8}; \frac{3}{8}\right]$	$\left[\frac{3}{8}; \frac{4}{8}\right]$	$\left[\frac{4}{8}; \frac{5}{8}\right]$	$\left[\frac{5}{8}; \frac{6}{8}\right]$	$\left[\frac{6}{8}; \frac{7}{8}\right]$	$\left[\frac{7}{8}; 1\right]$
1	2,667	2,180	1,354	0,639	0,229	0,062	0,013	$2,03 \cdot 10^{-3}$
2	2,686	2,363	1,579	0,802	0,309	0,091	0,020	$3,41 \cdot 10^{-3}$
3	2,654	2,512	1,806	0,987	0,410	0,129	0,031	$5,63 \cdot 10^{-3}$
4	2,571	2,618	2,026	1,191	0,532	0,180	0,046	$9,11 \cdot 10^{-3}$
5	2,442	2,677	2,229	1,410	0,677	0,247	0,068	$1,44 \cdot 10^{-2}$
6	2,275	2,683	2,404	1,636	0,846	0,332	0,099	$2,25 \cdot 10^{-2}$
7	2,079	2,638	2,543	1,862	1,036	0,438	0,141	$3,43 \cdot 10^{-2}$
8	1,862	2,543	2,638	2,077	1,244	0,566	0,196	$5,14 \cdot 10^{-2}$
9	1,636	2,404	2,683	2,275	1,466	0,717	0,267	$7,54 \cdot 10^{-2}$
10	1,410	2,229	2,677	2,442	1,693	0,892	0,357	$1,08 \cdot 10^{-2}$
11	1,191	2,026	2,618	2,571	1,918	1,087	0,468	$1,53 \cdot 10^{-1}$
12	0,987	1,806	2,512	2,654	2,130	1,299	0,602	0,212
13	0,802	1,579	2,363	2,686	2,320	1,522	0,759	0,287
14	0,639	1,354	2,180	2,667	2,478	1,750	0,939	0,382
15	0,499	1,138	1,972	2,596	2,596	1,972	1,138	0,499
16	0,382	0,939	1,750	2,478	2,667	2,180	1,354	0,639
17	0,287	0,759	1,522	2,320	2,686	2,363	1,579	0,802
18	0,212	0,602	1,299	2,130	2,654	2,512	1,806	0,987
19	0,153	0,468	1,087	1,918	2,571	2,618	2,026	1,191
20	0,108	0,357	0,892	1,693	2,442	2,677	2,229	1,410
21	$7,54 \cdot 10^{-2}$	0,267	0,717	1,466	2,275	2,683	2,404	1,636
22	$5,14 \cdot 10^{-2}$	0,196	0,566	1,244	2,077	2,638	2,543	1,862
23	$3,43 \cdot 10^{-2}$	0,141	0,438	1,036	1,862	2,543	2,638	2,079
24	$2,25 \cdot 10^{-2}$	0,099	0,332	0,846	1,636	2,404	2,683	2,275
25	$1,44 \cdot 10^{-2}$	0,068	0,247	0,677	1,410	2,229	2,677	2,442
26	$9,11 \cdot 10^{-3}$	0,046	0,180	0,532	1,191	2,026	2,618	2,571
27	$5,63 \cdot 10^{-3}$	0,031	0,129	0,410	0,987	1,806	2,512	2,654
28	$3,41 \cdot 10^{-3}$	0,020	0,091	0,309	0,802	1,579	2,363	2,686
29	$2,03 \cdot 10^{-3}$	0,013	0,062	0,229	0,639	1,354	2,180	2,667
30	$1,18 \cdot 10^{-3}$	$8,09 \cdot 10^{-3}$	0,042	0,166	0,499	1,138	1,972	2,596

Лабораторная работа 9

Значения определенного интеграла, вычисленные по формуле Симпсона при $2n = 6$.

Номер варианта	$\int_a^b f(x) dx$	Номер варианта	$\int_a^b f(x) dx$
1	1,111	16	1,174
2	6,230	17	2,152
3	3,059	18	3,104
4	0,659	19	3,104
5	0,086	20	0,587
6	1,118	21	1,250
7	1,124	22	0,526
8	8,980	23	0,364
9	0,310	24	-2,150
10	0,904	25	0,787
11	0,747	26	1,075
12	2,546	27	1,463
13	2,546	28	0,747
14	-0,123	29	2,754
15	0,875	30	2,020

Лабораторная работа 10

Номер варианта	x_{\min}	f_{\min}	Номер варианта	x_{\min}	f_{\min}
1	-0,35	0,827	16	0,35	0,827
2	0,70	3,442	17	-0,70	3,442
3	0,57	2,330	18	1,76	2,330
4	0,00	2,000	19	0,00	2,000
5	0,74	2,264	20	-0,74	2,264
6	-0,99	0,174	21	0,99	0,174
7	-0,45	-0,232	22	0,45	-0,232
8	0,01	1,000	23	-0,01	1,000
9	-0,38	-9,049	24	0,62	-5,508
10	-0,28	-0,133	25	0,28	-0,133
11	-0,99	2,022	26	0,99	2,022
12	2,02	3,442	27	0,50	3,442
13	0,70	0,827	28	1,42	0,827
14	1,76	2,330	29	1,76	2,330
15	0,91	2,264	30	1,00	2,000

Лабораторная работа 11

Номер варианта	x_{\min}	y_{\min}	f_{\min}	Номер варианта	x_{\min}	y_{\min}	f_{\min}
1	-0,40	-0,90	0,523	16	-0,28	-0,28	0,728
2	0,00	0,00	2,000	17	1,00	0,00	-1,000
3	0,35	0,00	0,827	18	0,00	1,10	-0,296
4	0,00	-0,35	0,827	19	0,00	-0,35	0,827
5	0,00	0,59	-0,277	20	-0,69	0,00	0,614
6	0,69	0,00	0,614	21	0,00	0,35	0,827
7	0,00	1,00	-1,000	22	-0,59	0,00	-0,277
8	1,10	0,00	-0,296	23	0,35	0,00	0,827
9	-0,28	0,28	0,728	24	0,00	-0,59	-0,277
10	0,00	0,00	2,000	25	0,28	0,28	0,728
11	-0,35	0,00	0,827	26	0,00	-1,00	-1,00
12	-0,35	0,00	0,827	27	-1,10	0,00	-0,296
13	0,35	0,00	0,827	28	0,00	0,69	0,614
14	0,65	0,00	-0,390	29	0,00	-1,10	-0,296
15	0,00	0,35	0,827	30	0,00	-0,69	0,614

Лабораторная работа 12

Номер варианта	$y(0)$	$y(0,02)$	$y(0,04)$	$y(0,06)$
1	1	1,02041	1,04169	1,06390
2	1	1,02061	1,04249	1,06571
3	0	0,02000	0,04001	0,06004
4	0	0,02020	0,04083	0,06191
5	0	0,02040	0,04163	0,06371
6	0	0,02020	0,04079	0,06176
7	0	0,02020	0,04084	0,06195
8	1	1,01712	1,03482	1,05311
9	1	1,02061	1,04249	1,06571
10	0	0,02040	0,04163	0,06371
11	0	0,02040	0,04163	0,06371
12	1	0,98039	0,96156	0,94347
13	0	$-2,67 \cdot 10^{-6}$	$-2,13 \cdot 10^{-7}$	$-7,20 \cdot 10^{-5}$
14	1	0,98059	0,96232	0,94513
15	0	0,02000	0,03997	0,05980
16	0	0,02020	0,04079	0,06176
17	0	-0,01960	-0,03843	-0,05650
18	0	-0,01960	-0,03919	-0,05817
19	0	-0,01980	-0,03920	-0,05820
20	1	0,98346	0,96750	0,95213
21	1	0,98059	0,96232	0,94513
22	0	-0,01960	-0,03843	-0,05650
23	0	0,02000	0,04001	0,06004
24	1	1,01702	1,03443	1,05219
25	3	3,02231	3,04530	3,06898
26	1	1,02082	1,04341	1,06793
27	1	0,98059	0,96232	0,94513
28	1	0,94726	0,89756	0,85063
29	1	1,02076	1,04308	1,06705
30	1	1,02061	1,04247	1,06563

Лабораторная работа 13

Номер варианта	Номер шага	$u_{31}^{(n)}$	$u_{21}^{(n)}$	$u_{11}^{(n)}$	$u_{32}^{(n)}$	$u_{22}^{(n)}$	$u_{12}^{(n)}$
1	0	-0,75	-1,75	-1,75	-4,25	-5,25	-5,25
	10	-2,000	-1,000	-2,000	-7,000	-8,000	-7,000
2	0	12,25	8,25	5,25	12,75	8,75	5,75
	10	14,000	7,000	2,000	17,000	8,000	1,000
3	0	25	13,75	7	33,5	22,25	15,5
	10	26,000	11,000	2,000	46,000	16,000	-2,000
4	0	3,25	2,5	1,75	4,25	3,5	2,75
	7	3,000	2,000	1,000	6,000	4,000	2,000
5	0	0,5	-1	-2,5	2,5	1	-0,5
	8	1,000	-2,000	-5,000	5,000	2,000	-1,000
6	0	2,75	3,5	4,25	1,75	2,5	3,25
	6	2,000	4,000	6,000	1,000	2,000	3,000
7	0	-1,75	-2,5	-3,25	-2,75	-3,5	-4,25
	10	-1,000	-2,000	-3,000	-2,000	-4,000	-6,000
8	0	2,5	1	-0,5	0,5	-1	-2,5
	10	5,000	2,000	-1,000	1,000	-2,000	-5,000
9	0	6,25	3,5	6,75	-2,75	-5,5	-2,25
	10	18,000	2,000	-2,000	-9,000	-16,000	-11,000
10	0	28,25	16,25	8,75	37,25	25,25	17,75
	12	28,828	12,900	2,961	51,410	19,814	-0,056
11	0	8	6,25	5,5	8,5	6,75	6
	10	9,286	5,857	3,286	9,286	6,857	5,286
12	0	6,25	5,25	5,25	7,75	6,75	6,75
	8	7,095	3,286	1,762	10,095	6,286	4,762
13	0	6	6	6	6	6	6
	8	5,000	6,000	7,000	7,000	6,000	5,000
14	0	3	1,5	1,5	4	2,5	2,5
	10	4,501	0,982	-0,165	5,023	-0,410	-1,644
15	0	4,25	3,5	2,75	3,25	2,5	1,75
	10	6,000	4,000	2,000	3,000	2,000	1,000

Ответы к вариантам 16—30 получим умножением на (-1) соответствующих результатов (варианты 1—15), увеличив на 15 номер варианта.

Лабораторная работа 14

$MU = 2,5 \cdot 10^{-1}$; $DX = 7,37 \cdot 10^{-3}$; $DY = 1,05 \cdot 10^{-2}$;
 $K(X, Y) = -2,15 \cdot 10^{-3}$; $\rho(X, Y) = -2,44 \cdot 10^{-1}$; $MX =$
 $= 1,77 \cdot 10^{-1} + 0,1n$ при $n = 1—15$; $MX = -1,377 \cdot 0,1n$
при $n = 16—30$.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Алберг Дж., Нильсон Э., Уолли Дж. Теория сплайнов и ее применение.— М. : Мир, 1972.— 316 с.
2. Бахвалов Н. С. Численные методы.— М. : Наука, 1975.— 630 с.
3. Белый Ю. А. Считывающая микроэлектроника.— М. : Наука, 1983.— 120 с.
4. Березин И. С., Жидков Н. П. Методы вычислений.— М. : Наука, 1966.— Т. 1.— 464 с.
5. Березин И. С., Жидков Н. П. Методы вычислений.— М. : Физматгиз, 1962.— Т. 2.— 620 с.
6. Васильев Ф. П. Лекции по методам решения оптимальных задач.— М. : Изд-во Моск. ун-та, 1974.— 374 с.
7. Васильева В. Н. Основы теории сплайнов.— Иркутск: Изд-во Иркут. ун-та, 1982.— 174 с.
8. Вергасов В. А., Журкин И. Г., Красиков М. В. Вычислительная математика.— М. : Недра, 1976.— 230 с.
9. Воеводин В. В. Вычислительные основы линейной алгебры.— М. : Наука, 1977.— 303 с.
10. Воеводин В. В., Кузнецов Ю. А. Матрицы и вычисления.— М. : Наука, 1984.— 320 с.
11. Волков Е. А. Численные методы.— М. : Наука, 1982.— 254 с.
12. Грабарь Л. П. Таблицы полиномов Чебышева, ортонормированных на системе равноотстоящих точек.— М. : Вычисл. центр АН СССР, 1965.— 198 с.
13. Гребенников А. И. Метод сплайнов и решение некорректных задач теории приближений.— М. : Изд-во Моск. ун-та, 1983.— 208 с.
14. Демидович Б. П., Марон И. А. Основы вычислительной математики.— М. : Наука, 1966.— 664 с.
15. Демидович Б. П., Марон И. А., Шувалова Э. З. Численные методы анализа.— М. : Наука, 1967.— 368 с.
16. Дьяконов В. П. Расчет нелинейных и импульсных устройств на программируемых микрокалькуляторах : Справ. пособие.— М. : Радио и связь, 1984.— 176 с.
17. Жалдак М. І., Рамський Ю. С. Програмування на мікрокалькуляторах : Посібник.— К. : Рад. шк., 1985.— 224 с.
18. Зав'ялов Ю. С., Красов Б. И., Мирошиниченко В. Л. Методы сплайн-функций.— М. : Наука, 1980.— 352 с.
19. Калиткин Н. Н. Численные методы.— Наука, 1978.— 512 с.
20. Канторович Л. В., Акилов Г. П. Функциональный анализ.— М. : Наука, 1977.— 740 с.
21. Коллатц Л. Функциональный анализ и вычислительная математика.— М. : Мир, 1969.— 447 с.

22. Копченова Н. В., Марон И. А. Вычислительная математика в примерах и задачах.— М. : Наука, 1972.— 367 с.
23. Корнейчук Н. П. Сплайны в теории приближения.— М. : Наука, 1984.— 352 с.
24. Ляшко И. И., Макаров В. Л., Скоробогатько А. А. Методы вычислений.— К. : Вища шк. Головное изд-во, 1977.— 406 с.
25. Макаров В. Л., Хлебыстов В. В. Сплайн-аппроксимация функций.— М. : Высш. шк., 1983.— 80 с.
26. Плис А. И., Сливина Н. А. Лабораторный практикум по высшей математике.— М. : Высш. шк., 1983.— 208 с.
27. Сигорский В. П. Математический аппарат инженера.— К. : Техника, 1977.— 768 с.
28. Стечкин С. Б., Субботин Ю. Н. Сплайны в вычислительной математике.— М. : Наука, 1976.— 248 с.
29. Трохименко Я. К., Любич Ф. Д. Инженерные расчеты на микрокалькуляторах.— К. : Техника, 1980.— 381 с.
30. Трохименко Я. К., Любич Ф. Д. Инженерные расчеты на программируемых микрокалькуляторах.— К. : Техника, 1985.— 328 с.
31. Трохименко Я. К., Любич Ф. Д. Радиотехнические расчеты на микрокалькуляторах.— М. : Радио и связь, 1983.— 256 с.
32. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры.— М. : Наука, 1960.— 656 с.
33. Форсайт Дж., Малькольм М., Моулер К. Машины методы математических вычислений.— М. : Мир, 1980.— 279 с.
34. Хемминг Р. В. Численные методы.— М. : Наука, 1968.— 398 с.
35. Химмельблау Д. Прикладное нелинейное программирование.— М. : Мир, 1975.— 536 с.
36. Цветков А. Н. Прикладные программы на микро-ЭВМ «Электроника Б3-21».— М. : Финансы и статистика, 1972.— 128 с.
37. Цветков А. Н., Епанечников В. А. Прикладные программы для микро-ЭВМ «Электроника Б3-34», «Электроника МК-56», «Электроника МК-54».— М. : Финансы и статистика, 1984.— 175 с.

О ГЛАВЛЕНИЕ

Предисловие	3
Глава 1. Программируемые электронные микрокалькуляторы	6
1.1. Краткая история развития вычислительных средств	6
1.2. Общие сведения о программируемых микрокалькуляторах	9
1.3. Ввод чисел	11
1.4. Микрокалькуляторы «Электроника Б3-21» и «Электроника МК-46»	14
1.5. Микрокалькуляторы «Электроника Б3-34», «Электроника МК-54», «Электроника МК-56», «Электроника МК-61» и «Электроника МК-52»	19
1.6. Работа микрокалькулятора в автоматическом режиме	32
1.7. Работа микрокалькулятора в программируемом режиме	33
1.8. Составление программы	34
1.9. Ввод исходных данных и пуск программы	38
1.10. Запись программы	40
Глава 2. Элементы теории погрешностей	42
2.1. Классификация погрешностей	42
2.2. Абсолютная и относительная погрешности	43
2.3. Запись приближенных чисел. Правила округления	44
2.4. Погрешности функций	45
2.5. Погрешности вычислений на микрокалькуляторах	48
Глава 3. Методы решения систем линейных алгебраических уравнений	50
3.1. Решение системы линейных алгебраических уравнений методом Гаусса	50
3.2. Вычисление определителя и обратной матрицы методом Гаусса	56
3.3. Погрешность решения системы линейных уравнений	57
3.4. Лабораторная работа 1. Решение системы линейных алгебраических уравнений методом Гаусса—Жордана	61
3.5. Метод прогонки	67
3.6. Итерационные методы решения линейных систем	70
Глава 4. Численные методы решения нелинейных уравнений и систем	77
4.1. Численные методы решения нелинейного уравнения с одним неизвестным	77
4.2. Лабораторная работа 2. Приближенное решение нелинейного уравнения методом Ньютона—Рафсона	87
4.3. Численные методы решения системы нелинейных уравнений	89
4.4. Лабораторная работа 3. Приближенное решение системы двух нелинейных уравнений методом простой итерации	92

4.5. Решение систем нелинейных уравнений методом Ньютона	97
4.6. Лабораторная работа 4. Приближенное решение системы двух нелинейных уравнений методом Ньютона	98
Глава 5. Приближение функций	102
5.1. Постановка задачи приближения функций	102
5.2. Полиномиальная интерполяция	102
5.3. Лабораторная работа 5. Полиномиальная интерполяция	107
5.4. Аппроксимация функции методом наименьших квадратов	119
5.5. Точечная аппроксимация функции методом наименьших квадратов	127
5.6. Точечная аппроксимация функций алгебраическими многочленами	130
5.7. Лабораторная работа 6. Точечная аппроксимация функции алгебраическими многочленами методом наименьших квадратов	133
5.8. Точечная аппроксимация функций ортогональными полиномами	135
5.9. Точечная аппроксимация функций тригонометрическими полиномами	139
5.10. Лабораторная работа 7. Аппроксимация функции тригонометрическими полиномами по методу наименьших квадратов	144
5.11. Интегральная аппроксимация функции методом наименьших квадратов	149
5.12. Лабораторная работа 8. Интегральная аппроксимация функции методом наименьших квадратов ортогональными полиномами	153
5.13. Аппроксимация сплайнами	159
5.14. Определение и способы аналитического представления сплайнов	164
5.15. Интерполяция кубическими сплайнами	166
5.16. Кубические В-сплайзы	172
Глава 6. Численное дифференцирование и интегрирование функций одной переменной	183
6.1. Численное дифференцирование функций с помощью интерполяционных полиномов	191
6.2. Численное дифференцирование функций с помощью сплайнов	198
6.3. Численное интегрирование функций одной переменной	201
6.4. Лабораторная работа 9. Численное интегрирование функции одной переменной	214
Глава 7. Численные методы минимизации функции	221
7.1. Постановка задачи	221
7.2. Одномерная минимизация функций	223
7.3. Лабораторная работа 10. Одномерная минимизация функции	225
7.4. Метод покоординатного спуска	227
7.5. Метод градиентного спуска	229
7.6. Метод Ньютона	230
7.7. Лабораторная работа 11. Методы минимизации функции нескольких переменных	233
Глава 8. Численные методы решения задачи Коши для обыкновенных дифференциальных уравнений	239
8.1. Метод Эйлера	239
8.2. Метод Рунге—Кутта	244

8.3. Метод Адамса	246
8.4. Метод Милна	248
8.5. Лабораторная работа 12. Численные методы решения задачи Коши	249
Глава 9. Решение краевых задач математической физики методом сеток	254
9.1. Метод сеток	254
9.2. Решение задачи Дирихле для уравнения Лапласа методом сеток	256
9.3. Решение первой краевой задачи для уравнения параболического типа методом сеток	258
9.4. Решение первой краевой задачи для уравнения гиперболического типа методом сеток	260
9.5. Лабораторная работа 13. Решение задачи Дирихле для уравнения Лапласа в прямоугольной области методом сеток	261
Глава 10. Вычисление числовых характеристик случайных векторов	265
10.1. Аксиоматическое определение вероятности	265
10.2. Случайные величины и случайные векторы	266
10.3. Числовые характеристики случайного двумерного вектора	267
10.4. Лабораторная работа 14. Вычисление числовых характеристик двумерного случайного вектора	268
<i>Ответы к заданиям по лабораторным работам</i>	272
<i>Список рекомендуемой литературы</i>	284

Учебное пособие

Галина Григорьевна Барановская
Ирина Николаевна Любченко

МИКРОКАЛЬКУЛЯТОРЫ В КУРСЕ ВЫСШЕЙ МАТЕМАТИКИ

Практикум

Редактор Л. И. Мубаракшина
Художественный редактор С. П. Духленко
Технический редактор А. И. Омоховская
Корректор А. Г. Телалова

Информ. бланк № 10837

Сдано в набор 27.06.86. Подп. в печать 09.01.87. БФ 26547. Формат 84×108^{1/32}. Бумага типогр. № 1. Лит. гарн. Выс. печать. Усл. печ. л. 15.12. Усл. кр.-отт. 15.38. Уч.-изд. л. 15.5. Тираж 20000 экз. Изд. № 7515. Зак. 6-467. Цена 70 к.

Головное издательство издательского объединения «Вища школа», 252054, Киев-54, ул. Гоголевская, 7.

Книжная фабрика им. М. В. Фрунзе, 310057, Харьков-57, Донец-Захаржевского, 6/8.