Л. Ю. Астанин, Ю. Д. Дорский, А. А. Костылев

Применение программируемых калькуляторов для инженерных и научных расчетов

Ленинград
Ленинградское отделение
ЭНЕРГОАТОМИЗДАТ
1986

ББК 32.97
А 91
УДК 681.32.181.48

Рецензенты *В. Б. Смолов, В. Д. Байков*

Астанин Л. Ю. и др.

А 91 Применение программируемых калькуляторов для инженерных и научных расчетов/Л. Ю. Астанин, Ю. Д. Дорский, А. А. Костылев. — Л.: Энергоатомиздат. Ленингр. отд-ние, 1986. — 176 с.: ил.

В книге описываются программируемые микрокалькуляторы «Электроника Б3-21» и «Электроника Б3-34», излагаются основы их использования. Рассматриваются вычислительные методы, широко применяемые в физике, автоматике, электротехнике и электронике, в соответствии с которыми приведены программы анализа функций, решения уравнений, статистического анализа и моделирования случайных величин и процессов. Наличие справочных материалов позволяет решать все рассмотренные задачи без обращения к другим источникам.

Книга предназначена для инженерно-технических работников и специалистов, связанных с вычислительными работами.

А $\frac{2405000000-121}{051(01)-86}$ 276—86 ББК 32.97

ПРОИЗВОДСТВЕННОЕ ИЗДАНИЕ

Лев Юрьевич Астанин
Юрий Дмитриевич Дорский
Александр Александрович Костылев

Применение программируемых калькуляторов для инженерных и научных расчетов

Редактор *С. П. Левкович*
Художественный редактор *Д. Р. Стеванович*
Технический редактор *Н. А. Минеева*
Корректор *Н. Б. Чухутина*
Обложка художника *И. М. Отрощенко*

ИБ № 1422

Сдано в набор 09.01.86. Подписано в печать 15.05.86. М-32939. Формат $84\times108^1/_{32}$. Бумага типографская № 2. Гарнитура литературная. Высокая печать. Усл. печ. л. 9,24. Усл. кр.-отт. 9,56. Уч.-изд. л. 10,62. Тираж 100 000 экз. Заказ 442. Цена 55 к.
Ленинградское отделение Энергоатомиздата. 191065, Ленинград, Марсово поле, 1.

Владимирская типография Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли 600000, г. Владимир, Октябрьский пр., д. 7.

© Энергоатомиздат, 1986

ПРЕДИСЛОВИЕ

Появление электронных вычислительных машин справедливо рассматривалось как качественно новый этап развития производительных сил. Благодаря широкому внедрению ЭВМ одним из основных методов современной науки и инженерной практики стал вычислительный эксперимент [24], позволяющий создавать и изучать математические модели различных объектов и явлений.

Создание персональных микро-ЭВМ поставило перед специалистами любых профессий задачу оптимального использования таких вычислительных средств. Наличие существенно общих черт в конструкции и методах использования программируемых вычислительных средств любых классов упрощает работу с ними.

Широкие возможности для изучения общих принципов использования ЭВМ, в том числе при первоначальном обучении, открываются при использовании цифровых вычислительных устройств индивидуального пользования — микрокалькуляторов (МК). Решение задач с помощью программируемых МК и ЭВМ имеет одинаковые этапы. Приобретение навыков работы с МК и максимальное использование их возможностей являются одним из эффективных путей овладения методами работы на любых ЭВМ. Индивидуальный и непрерывный характер общения пользователя с МК, возможность в любой момент времени получить сведения о данных, программе и внести необходимые изменения обусловливает быстрое решение любых поставленных задач, что является бесспорным преимуществом МК. Конечно, современные мини- и микро-ЭВМ с дисплеями предоставляют не меньшие возможности для диалога оператора и ЭВМ, однако пока они относятся к дефицитному оборудованию.

Вычислительные возможности МК позволяют решать достаточно сложные инженерные и научные задачи, причем с меньшими суммарными затратами времени, чем на больших ЭВМ.

Книга состоит из двух разделов. В первом разделе на примерах отечественных микрокалькуляторов «Электроника Б3-21» и «Электроника-Б3-34» рассматриваются общие принципы построения и использования программируемых МК. При этом авторы стремились подчеркнуть общность структуры, методов программирования и использования МК и ЭВМ других типов.

Во втором разделе рассматриваются вопросы практического применения программируемых МК. Представленные прикладные программы охватывают основные области применения вычислительных средств: анализ функций и решение уравнений, статистический анализ и моделирование случайных величин и случайных последовательностей. Для лучшего понимания перед каждой программой приводятся краткие теоретические сведения, а в приложения вынесены необходимые статистические таблицы. Это позволяет решать все задачи без обращения к другим источникам. Представленные программы существенно дополняют известные сборники прикладных программ [8, 31, 38], особенно в области статистического анализа, и служат хорошей иллюстрацией основных положений и рекомендаций, развиваемых в первой части книги.

Отзывы и пожелания просьба направлять по адресу: 191065, Ленинград, Марсово поле, д. 1, Ленинградское отделение Энергоатомиздата.

Авторы

РАЗДЕЛ ПЕРВЫЙ

АРХИТЕКТУРА, ТЕХНИЧЕСКИЕ ПОКАЗАТЕЛИ И ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ МИКРОКАЛЬКУЛЯТОРОВ

Глава первая

ПРИНЦИПЫ ПОСТРОЕНИЯ И ТЕХНИЧЕСКИЕ ПОКАЗАТЕЛИ МИКРОКАЛЬКУЛЯТОРОВ. ОСНОВНЫЕ ПОНЯТИЯ ПРОГРАММИРОВАНИЯ

1.1. Общие сведения о микрокалькуляторах

Функциональные возможности МК. Электронные клавишные вычислительные машины индивидуального пользования, или микрокалькуляторы (МК), предназначены для выполнения расчетов при решении сложных инженерных и научных задач. Расчеты, даже очень простые, проводятся по программе. Программа определяет последовательность операций над числами (операндами), вводимыми в начале вычислений или получаемыми в результате предыдущих операций. В МК операции выполняются по командам, подаваемым оператором при нажатии на соответствующие клавиши, или автоматически в соответствии с введенной программой. Символы выполняемых операций изображены на клавиатуре, их совокупность образует алфавит языка программирования МК. Перечень правил, по которым осуществляется ввод данных и команд, образует грамматику языка МК. Последовательность вводимых символов составляет программу вычислений, отдельные части которой на входном языке называют предложениями.

К настоящему времени существует уже несколько поколений МК, отличающихся своими функциональными и техническими показателями, языком и грамматикой [18].

Обычно МК классифицируют по их функциональным возможностям. Простейшие калькуляторы, появившиеся на первом этапе развития микроэлектроники, выполняют четыре арифметических действия и несколько других элементарных операций, например нахождение обратного значения.

В составе языка МК для инженерных расчетов имеются команды, вызывающие выполнение достаточно сложных вычислений, например, значений стандартных функций — тригонометрических, логарифмических, степенных или статистических характеристик случайных величин — математического ожидания, дисперсии и др. Такие функции вычисляются по микропрограммам. Микропрограммы в МК хранятся в специальном постоянном запоминающем устройстве (ПЗУ). Обращение к ним выполняется при нажатии соответствующей клавиши, а время вычислений не превышает долей или единиц секунд. Программа сложных расчетов реализуется путем последовательного нажатия клавиш оператором.

Наиболее значительным достижением в развитии функциональных возможностей МК явилось создание программируемых МК, способных запоминать вводимую оператором программу вычислений и затем выполнять ее в автоматическом режиме. Для запоминания программы предусматривается оперативное запоминающее устройство (ОЗУ). Последовательность команд, записанных в ОЗУ, выполняется автоматически, таким образом, использование ОЗУ заменяет нажатие оператором клавиш МК. На первый взгляд может показаться, что имеет место простое улучшение эксплуатационных характеристик устройства. Однако речь идет о качественном отличии программируемых МК. Часто выполняемые с их помощью расчеты почти невозможно реализовать простым нажатием клавиш как из-за большой продолжительности этих расчетов, так и вследствие неизбежных при этом ошибок в действиях оператора. Введение программы в ОЗУ может быть проверено и скорректировано, что полностью исключает подобные ошибки. Последнее позволяет рассматривать программируемые МК как полноправных представителей ЭВМ.

В настоящей книге рассматриваются отечественные программируемые МК типов «Электроника Б3-21» и «Электроника Б3-34» (в дальнейшем будем писать Б3-21 и Б3-34). Они характеризуются емкостью памяти, равной 60 и 98 шагам программы, наличием соответственно семи и четырнадцати регистров памяти для хранения исходных данных и промежуточных результатов. Такие показатели в значительной мере ограничивают возможности применения МК. В перспективных моделях (например МК-61) предполагается увеличение емкости ОЗУ и числа регистров памяти, а также расширение автоматически выполняемых операций. Очень важно также сохранять программу при выключении питания МК, и в этом достигнуты существенные успехи (МК-52). Использование устройств для регистрации результатов вычислений и построения графиков, а также сопряжение МК с внешними запоминающими устройствами значительно сблизят возможности МК и ЭВМ [18].

В отдельную группу выделяют специализированные МК, функциональное назначение которых чрезвычайно разнообразно. Кроме использования в качестве часов, календаря, будильника, появляются МК, выполняющие роль словаря-переводчика. Широкое распространение получают электронные игры.

Эксплуатационные показатели МК. К эксплуатационным показателям МК относятся надежность, ремонтопригодность, продолжительность непрерывной работы и т. д. Один из наиболее существенных показателей — потребляемая мощность источника питания. Можно выделить две группы МК: первая имеет потребляемую мощность десятки и сотни милливатт, вторая — менее 1 милливатта. При работе от гальванических батарей или аккумуляторов МК первой группы сохраняют работоспособность в течение единиц или десятка часов, т. е. необходимо использовать преобразователь напряжения сети переменного тока.

Для МК с мощностью менее 1 милливатта время работы от автономных источников достигает тысяч часов. Так, простейший калькулятор МК-53 потребляет мощность 0,06 мВт, что обеспечивает автономную работу от двух элементов СЦ-32 в течение восьми тысяч часов.

Отметим, что мощность потребления МК непосредственно зависит от типа используемого индикатора. Например, в МК, мощность потребления которого меньше 1 мВт, применяется лишь жидкокристаллический индикатор, широко используемый в наручных электронных часах. Жидкокристаллический индикатор потребляет очень малую мощность, однако, к сожалению, при пониженной температуре яркость свечения падает. В более энергоемких МК используются светодиодные и катодолюминесцентные индикаторы.

Логическая организация МК. Важнейшей характеристикой МК является используемая в них грамматика, т. е. правила ввода данных и последовательность выполняемых операций, или иначе их логическая структура. По логической структуре различают МК с алгебраической логикой, с алгебраической логикой с иерархией операций, с алгебраической логикой со скобками и с инверсной польской записью [18, 40].

В МК с алгебраической логикой операции выполняются в том порядке, в котором они вводятся. Организация более сложных вычислений в таких МК требует регистрации вручную или в специальной памяти результатов промежуточных этапов. В МК с алгебраической логикой и иерархией операций правильный порядок выполнения действий осуществляется автоматически. Структура таких МК должна быть усложнена введением дополнительной памяти и логических устройств управления ею, а максимальная сложность вычисляемых выражений ограничена используемым объемом этого дополнительного оборудования.

У МК с алгебраической логикой со скобками в число символов языка введены открывающие и закрывающие скобки, используя которые оператор определяет необходимый порядок выполнения операций. Логика со скобками в наибольшей степени совпадает с привычным способом записи алгебраических выражений. Успехи микроэлектроники обеспечили производство МК с двумя-тремя уровнями скобок. Заметим, что логика со скобками означает использование системы, в которой выполнение некоторых операций откладывается на тем большее число шагов, чем больше уровень скобок. Такая процедура обычно реализуется с помощью запоминающего устройства со специальной дисциплиной обмена, называемого стеком (см. § 1.2).

Кроме рассмотренных вариантов алгебраической логики в МК широко применяется логическая организация, называемая инверсной польской логикой [18], которая позволяет получить однозначную запись арифметических выражений без использования иерархии операций и скобок. Термин «инверсная» характеризует принятый способ записи операций: сначала записываются операнды, а затем символ операции. Операции производятся в прямом порядке записи их символов над предшествующим операндом при одноместной операции или над двумя предшествующими — при двухместной операции. Результат операции запоминается как новый операнд для последующей операции. Дополнительная память МК при

инверсной польской логике невелика, поскольку требуется запоминание лишь двух операндов для выполняемой операции и результата операции как операнда для последующей. Команды вводятся вручную или из программной памяти в порядке их выполнения. Это составляет заметное преимущество инверсной польской логики (по сравнению с алгебраической) и объясняет широкое ее распространение. Заметим, что отличительной чертой МК с инверсной польской записью является наличие клавиши «↑», используемой для разделения последовательно вводимых операндов.

1.2. Функциональная схема микрокалькулятора

Основные функциональные блоки МК. Для эффективной работы с МК необходимо ознакомиться с их структурой, функциональным назначением отдельных узлов и их взаимодействием в процессе вычислений. В наиболее общей форме структура МК (рис. 1.1) включает в себя операционное устройство (ОУ), устройство управления (УУ), память и устройства ввода — вывода. К такой схеме могут быть сведены ЭВМ любых классов, что и отражает отмечавшуюся типичность структуры МК. В ОУ обрабатываются данные, представленные в цифровой форме. В состав операций ОУ входят два-три десятка простейших арифметических и логических операций. Возможность выполнения сколь угодно сложных операций обработки данных с помощью последовательного выполнения простейших операций доказана теоретически [2]. Простейшие операции, выполняемые ОУ, называются микрооперациями, а их последовательность, вызывающая выполнение сложной операции из состава входного языка МК,— микропрограммой. Микрооперации выполняются в ОУ под воздействием микрокоманд, формируемых в УУ. Для формирования микропрограмм в УУ используются специальные средства — память микропрограмм.

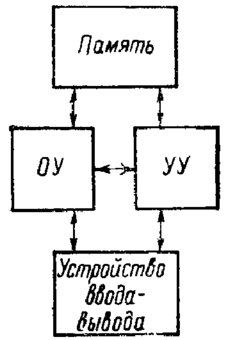


Рис. 1.1. Структурная схема МК

Память МК, как и ЭВМ любых типов, является принципиально необходимым элементом. Для пользователя МК необходимость памяти становится очевидной уже при выполнении операции сложения двух операндов, для чего нужно ввести операнды порознь, т. е. предварительно запомнить хотя бы один из них. При вычислении сложных арифметических выражений память требуется для запоминания исходных данных и промежуточных результатов. В программируемых МК память необходима и для записи последовательности команд, реализующих разработанные пользователем программы вычислений.

Обобщенная функциональная схема МК. Обобщенная функциональная схема программируемого МК (рис. 1.2) рассматривается применительно к отечественным МК типов Б3-21 и Б3-34.

В ОУ выполняются арифметические операции над одним или двумя операндами. С этой целью в МК предусмотрены два операционных регистра: РХ и РY. Число x, записанное в РХ, может использоваться различным образом. Во-первых, как операнд для проведения операции над величиной x, например, для вычисления x^2, $\ln x$, $\sin x$ и т. д. Такие операции называют одноместными. Во-вторых, число x может быть одним из двух операндов при двухместных операциях «+»; «—»; «×»; «:» и операции x^y. В-третьих, предварительная запись числа в РХ используется для последующей пересылки его в один из внутренних регистров памяти (РП) МК.

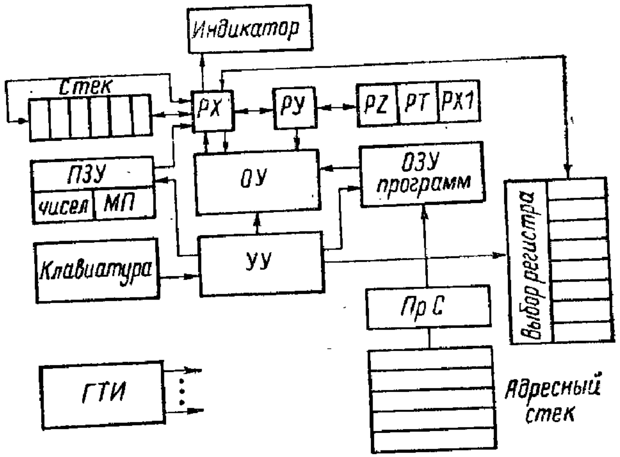


Рис. 1.2. Функциональная схема программируемого МК

При выполнении одноместной операции после записи числа x в РХ с помощью соответствующей клавиши формируется код требуемой команды, который воздействует на ОУ, обеспечивая выполнение требуемой операции над величиной x. Результат выполнения операции помещается в РХ. Регистры в ЭВМ с режимом предварительного хранения операнда и последующей записью результата называются регистрами-аккумуляторами [2]. Содержание РХ в форме десятичного числа высвечивается на индикаторе МК. Если МК выполняет последовательность операций в автоматическом режиме, на индикаторе наблюдается мерцание изображения.

Подготовка исходных данных для осуществления двухместной операции производится в два этапа: вначале операнд y записывается в РХ, затем специальной командой ↑ пересылается во второй операционный регистр РУ, а затем операнд x помещается в РХ и формируется команда на выполнение операции. Результат операции также помещается в РХ.

Устройством ввода МК является его клавиатура. С помощью клавиш осуществляется ввод чисел и команд для выполнения операций. Коды вызываемых цифр, а также микропрограммы хранятся в постоянном запоминающем устройстве (ПЗУ). Обращение к нуж-

ным ячейкам памяти ПЗУ при нажатии клавиши осуществляется с помощью устройства управления (УУ).

Виды устройств памяти в схеме МК. Стеки. Организация процесса вычислений достаточно сложных арифметических выражений связана с необходимостью запоминания исходных данных и промежуточных результатов. Наиболее очевидным, но не оптимальным способом является использование отдельных регистров памяти (РП). Такие регистры обычно объединяются в блок РП (см. рис. 1.2), а выбор нужного регистра осуществляется выработкой его адреса в УУ. Программирование решения задачи, т. е. организация процесса вычислений, должно предусматривать обмен данными блока РП и операционного регистра РХ.

Целесообразно отметить, что проблема размещения и передачи данных в различных регистрах МК, с одной стороны, представляет собой предмет наибольшего внимания при составлении программы, а с другой стороны, проверка содержимого РП является надежным средством контроля правильности вычислений при отладке программы. Рассмотрим, например, способы получения необходимых при вычислениях констант. Константа может быть введена в РП перед началом работы, а затем использоваться при выполнении вычислений по программе. Другой путь состоит в том, что восьмиразрядное число может быть вызвано в РХ из ПЗУ чисел, для чего может понадобиться до девяти шагов программы (считая вызов запятой). В общем случае альтернативное решение должно приниматься с учетом того, какие ограничения при составлении программы являются более жесткими: на число шагов программы или на число требуемых РП. Известны модели МК модульной структуры (*TI*-59), в которых возможно одни и те же элементы памяти использовать либо для записи программы, либо в качестве РП.

Усложнение логической структуры МК неизбежно связано с увеличением объема аппаратуры. Успехи микроэлектронной технологии ни в какой мере не снижают требований по минимизации объема этих затрат. Одним из наиболее эффективных путей разрешения противоречия является построение аппаратуры на однородных структурах с регулярными связями. Упоминавшийся ранее принцип микропрограммной реализации достаточно сложных алгоритмов, необходимых для выполнения команд в МК, основан на использовании узлов аппаратуры с однородной структурой, а именно ПЗУ. Для выполнения сложной операции устройство управления (УУ) лишь указывает адрес, т. е. номер ячейки памяти ПЗУ микропрограмм, в которой находится начальная микрокоманда микропрограммы. В дальнейшем микропрограмма выполняется как последовательность микроопераций, заданная программированием ПЗУ при его изготовлении. Таким образом, однократная команда УУ вызывает последовательность действий ОУ, что, естественно, способствует уменьшению объема аппаратуры УУ.

При вычислении сложных арифметических выражений алгоритм логики со скобками должен обеспечить последовательное вычисление выражений, записанных во внутренних скобках, запоминание результата, затем вычисление выражений в следующих скобках, т. е. требуется многоступенчатое «откладывание» предыдущего результата. Аналогично при использовании польской инверсной логики предварительно полученные результаты являются операндами для последующих операций в строгом порядке их выполнения по программе (см. § 1.1). Реализация такой задачи облегчается при использовании специального устройства памяти, получившего наименование стек. Стек представляет собой набор регистров (рис. 1.3), запись и чтение данных в котором осуществляется по определенному правилу: каждое новое число записывается в регистр с наименьшим номером (*P0*), а ранее записанные числа «сдвигаются» в регистры с большими номерами. Чтение осуществляется в обратном порядке: после считывания из *P0* происходит сдвиг данных: из *P1* в *P0*, из *P2* в *P1* и т. д. Такая организация памяти удобна при вычислениях с усложненной логикой. Для обеспечения возможности считывания данных из регистра с наибольшим номером стек может быть замкнут в кольцо. Перемещение данных в кольцевом стеке возможно в обоих направлениях. Следует отметить, что при обращении к стеку не происходит перезапись информации во всех его регистрах: это привело бы к неоправданному увеличению аппаратурных, временных, энергетических и других затрат. Фактически обращение всегда происходит к новому регистру, которому присваивается нулевой номер, а остальным регистрам присваиваются соответствующие возрастающие номера.

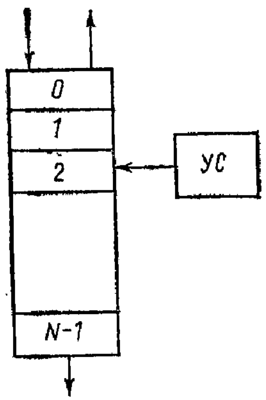


Рис. 1.3. Стек с магазинной организацией

«Перенумерация» регистров осуществляется с помощью специального счетчика (см. рис. 1.3), называемого указателем вершины стека (УС). Показания УС увеличиваются на единицу при каждой новой записи в стек и уменьшаются на единицу при каждом считывании. Таким образом, в УС всегда находится число, равное наибольшему номеру занятого регистра стека.

В рассматриваемой обобщенной структурной схеме МК имеется несколько стеков, предназначенных для различных целей. На рис. 1.2 показан кольцевой стек, замкнутый через операционный регистр РХ. Такой стек может быть доступен пользователю (Б3-21), тогда во входном языке МК имеются команды сдвига данных в стеке, включая которые в программу, можно организовать вычисление сложных арифметических выражений.

В МК с инверсной польской логикой строго регламентировано использование результата предыдущей операции в качестве операнда для последующей (см. § 1.1). При этом программирование вычисления арифметических выражений значительно облегчается, если в логической организации МК предусмотрено автоматическое перемещение результата операции с целью его сохранения и дальнейшего использования. Для этого в структуре МК предусматривается операционный стек, включающий в себя оба операционных регистра РХ и PY (Б3-21) и дополнительные регистры PZ и PT (Б3-34). При использовании такого стека результат выполнения предыдущей операции (содержание РХ) при записи нового числа в РХ автоматически переписывается в PY, что исключает специальную команду пересылки. Аналогично используются PZ и PT (см. гл. 2). При программировании в инверсной польской записи для сохранения результата предыдущей операции, который оказывается операндом для последующей, в составе операционного стека используется

дополнительный регистр РХ1 — регистр предыдущего результата (Б3-34).

Организация вычислений в автоматическом режиме. Рассмотренные элементы структуры в равной степени принадлежат МК с ручным управлением и пограммируемым МК. Работа МК в автоматическом режиме отличается тем, что последовательность команд выполняемой программы предварительно размещается в ОЗУ (см. рис. 1.2), допускающем запись программы, ее контроль, редактирование и автоматическое считывание по мере выполнения предыдущих команд. Для записи или контроля программы МК переводится в режим «Программирование». В этом режиме индикатор выполняет функцию дисплея, на котором высвечивается участок программы, состоящий из трех двузначных кодов последовательно записанных команд, а также номер (адрес) ячейки памяти ОЗУ, в которую занесена или будет занесена следующая за наблюдаемыми команда (см. § 2.1).

При нажатии клавиши, соответствующей требуемой операции, код этой операции записывается в очередную ячейку ОЗУ. Порядок следования адресов ОЗУ, т. е. порядок выполнения операций по программе, задается специальным элементом структуры программируемого МК (см. рис. 1.2) — программным счетчиком (ПрС). При нажатии клавиши МК, соответствующей очередной команде программы, код этой команды записывается в ячейку памяти ОЗУ, указанную ПрС, и отражается на крайних левых позициях индикатора. Коды предыдущих операций смещаются вправо, а показания ПрС изменяются, в простейшем случае — увеличиваются на единицу. В режиме «Работа» программируемого МК последовательность выполнения операций определяется изменением содержания ПрС. Обычно в программируемых МК предусматривается операция пошагового прохождения программы, тогда содержание ПрС изменяется при нажатии специальной клавиши. В автоматическом режиме выполнения программы содержание ПрС изменяется после выполнения предыдущей операции. Таким образом, ПрС является основным элементом, определяющим порядок вычислений в программируемых МК. Качественное отличие программируемых МК состоит в возможности автоматической оценки результатов вычислений и соответствующего изменения порядка выполнения программы: ее ветвления, многократного циклического повторения ее участков или перехода к подпрограммам. Могут использоваться и подпрограммы внутри подпрограмм. Аппаратурное решение подобной задачи достигается с помощью устройства памяти, организованного по принципу стека. Такой адресный стек, работающий с ПрС, изображен на рис. 1.2. Здесь же изображен генератор тактовых импульсов (ГТИ), синхронизирующий работу всех элементов МК.

Все элементы структурной схемы МК реализуются в микроэлектронном исполнении как большие интегральные схемы (БИС).

1.3. Основные понятия программирования

Алгоритм. При решении вычислительных задач любых классов после содержательного описания и построения математической модели требуется приведение ее к виду, пригодному для решения с помощью конкретной ЭВМ. Для этого выбирают алгоритм вычислений и составляют программы решения задачи. Оба эти различные этапа обычно объединяются при программировании решения задачи.

Под алгоритмом понимают точное предписание, определяющее содержание и порядок действий над исходными данными и промежуточными результатами, необходимыми для получения конечного результата при решении рассматриваемой задачи. Разработка алгоритмов вычислений составляет предмет обширной области знания — вычислительной математики, применение которой к математическим моделям объектов различной физической природы является процессом творческим, требующим специальных знаний. Существуют разнообразные методы решения типовых математических задач [17, 13]. Поэтому возможно построение различных алгоритмов решения одной и той же задачи.

Любой алгоритм должен обладать свойствами результативности, определенности и массовости. Результативность алгоритма означает возможность получения результата после конечного числа операций. Определенность состоит в совпадении окончательных и промежуточных результатов вне зависимости от того, кто ведет вычисления и какие технические средства используются. Массовость — есть возможность применения алгоритма в определенной области задания исходных данных. Выход из этой области может нарушить условия результативности алгоритма: вычисления никогда не окончатся [1].

Среди возможных алгоритмов решения задачи, удовлетворяющих перечисленным требованиям, выбор оптимального может производиться по различным частным критериям. Наиболее существенными из них являются время выполнения вычислений, объем описания алгоритма, точность вычислений и пригодность алгоритма для реализации решения с помощью конкретной ЭВМ. Оптимизация алгоритма по частным критериям обычно приводит к противоречивым ситуациям, разрешение которых также значительно зависит от разработчика.

Для записи разработанных алгоритмов используют словесную формулировку, графическое представление в виде схемы, представление в виде таблиц и др. Наибольшей простотой и наглядностью отличается графическое представление алгоритма в виде схемы, состоящей из ряда символов — блоков, каждому из которых соответствует определенный шаг решения задачи (ГОСТ 19.001—80).

Основные операции по реализации алгоритма выполняются арифметическими и логическими блоками. Последовательность выполнения алгоритма устанавливается с помощью упорядоченного размещения блоков на схеме и объединения их направленными линиями. Основные направления потока информации на схеме: сверху вниз и слева направо. В других случаях направление линий отмечается стрелками. Число линий, входящих в блок, не ограничивается, выходящая линия только одна. Исключение составляют логические блоки, имеющие не менее двух выходящих линий, каждая из которых соответствует одному из исходов проверки логического условия и соответственно определяет один из возможных путей продолжения вычислений. Внутри символов блоков записывают выполняемую операцию или логическое условие.

Требования к степени детализации схемы алгоритма противоречивы. Алгоритмы любой сложности можно представить в виде одного блока вычислений, однако такая схема не информативна. В то же время излишняя детализация сложного алгоритма достаточно быстро приводит к потере ее наглядности. Схема алгоритма, включающая в себя только операции из состава языка МК, является, по существу, схемой программы вычислений. Большая детализация схемы никогда не требуется, однако использование схемы с укрупненными блоками может быть весьма полезно на стадии разработки алгоритма и поблочной отладки программы.

Виды алгоритмов. Процесс разработки алгоритмов почти не поддается формализации вследствие многообразия решаемых задач. Поэтому большое практическое значение имеет поиск путей придания этому процессу систематического характера. Основой для этого является классификация алгоритмов по способу управления порядком выполнения действий. Существует всего три группы алгоритмов, различающихся по этому признаку: линейные, разветвляющиеся и циклические. Изучение методов представления сложных алгоритмов с помощью элементарных блоков трех названных типов относится к области структурированного программирования [1, 22]. Рассмотрим названные группы алгоритмов.

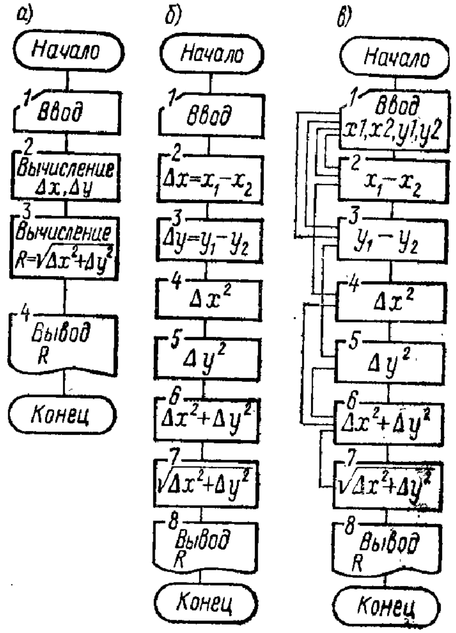


Рис. 1.4. Схемы линейных алгоритмов

Линейные алгоритмы предусматривают получение результата при однократном выполнении одной и той же последовательности действий для любых значений исходных данных. Схема линейного алгоритма представляет собой последовательность блоков вычислений с единственным входом каждый. Рассмотрим, например, задачу вычисления расстояния R между двумя точками (x_1, y_1) и (x_2, y_2) в декартовых координатах: $R=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$. Вводя промежуточные переменные $\Delta x=x_1-x_2$ и $\Delta y=y_1-y_2$, можно записать: $R=\sqrt{\Delta x^2+\Delta y^2}$ и представить схему линейного алгоритма в виде рис. 1.4, *а*. Ясно, что эта схема пригодна для ЭВМ, в составе языка которой имеется команда, реализующая операцию $\sqrt{a^2+b^2}$. В составе языка МК обычно имеются лишь команды «+», «—», «x^2» и «$\sqrt{x}$». Поэтому схема алгоритма вычислений для МК подобна схеме, приведенной на рис. 1.4, *б*. Такая схема может рассматриваться как схема программы МК, однако полностью процесс вычислений эта схема не отражает. Это объясняется тем, что в блоках вычислений *2—7* используются в качестве входных данных исходные данные задачи или результаты промежуточных вычислений. Для хранения этих данных требуются регистры памяти, а в состав программы должны быть введены команды записи в регистры и чтения из них. Передачу данных между блоками вычислений изображают с помощью линий (рис. 1,4, *в*), называемых информационными связями [9]. Каждая информационная связь означает, что регистр памяти занят и необходимо использовать соответствующую команду в программе. Простой рассматриваемый пример позволяет понять, что последовательность блоков в схеме алгоритма может быть различной, например, после вычисления Δx можно сразу вычислить Δx^2 и т. п. Сохраняя алгоритм вычислений, такие преобразования изме-

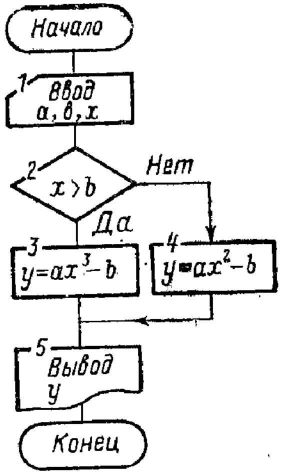


Рис. 1.5. Схема разветвляющегося алгоритма

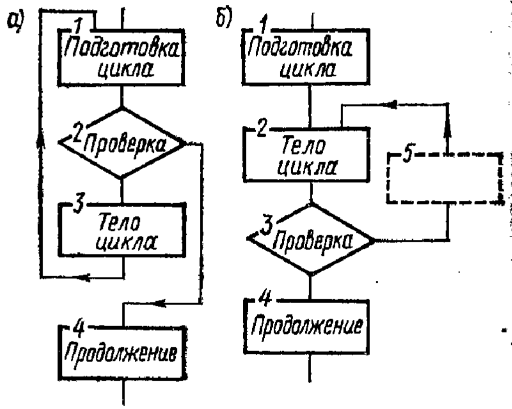


Рис. 1.6. Схемы циклических алгоритмов

няют информационные связи в программе и позволяют таким образом оптимизировать программу, сокращая требования к емкости памяти ЭВМ. В МК число регистров памяти невелико, поэтому подобные преобразования могут быть необходимым условием реализации вычислений.

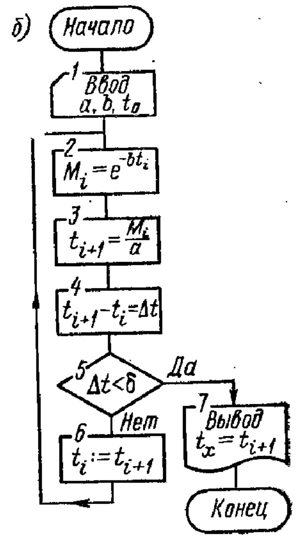
Разветвляющиеся алгоритмы предусматривают выбор одной из нескольких возможных последовательностей действий в зависимости от значений исходных данных или промежуточных результатов. Разветвляющийся алгоритм содержит по крайней мере один логический блок. Например, вычисление выражения

$$y = \begin{cases} ax^3 - b, & \text{если } x > b; \\ ax^2 - b, & \text{если } x \leqslant b \end{cases}$$

можно произвести по алгоритму, схема которого использует логический блок, проверяющий условие $x > b$ (рис. 1.5).

Наиболее сложными по структуре являются циклические алгоритмы, обеспечивающие получение результата с помощью многократного повторения некоторой последовательности действий. Блок повторяющихся вычислений алгоритма называют телом цикла. Число повторений цикла определяется некоторыми условиями, для про-

верки которых в циклический алгоритм включается логический блок. В циклическом алгоритме с верхним окончанием (рис. 1.6, *а*) условия проверяются до вычисления тела цикла. Может возникнуть ситуация, когда операции цикла не выполняются ни разу. Для циклического алгоритма с нижним окончанием (рис. 1.6, *б*) вычисления тела цикла выполняются минимум один раз. По числу повторений различают алгоритмы с заданным числом повторений (детерминированные) и итерационные. В первом случае в блоке подготовки цикла в специальный счетчик записывается требуемое число повторений и после каждого вычисления тела цикла содержание счетчика уменьшается или увеличивается на единицу вплоть до требуемого значения. В циклических алгоритмах итерационного типа выход из цикла осуществляется при выполнении некоторого условия, связанного с проверкой вычисляемой величины. Эта провер-



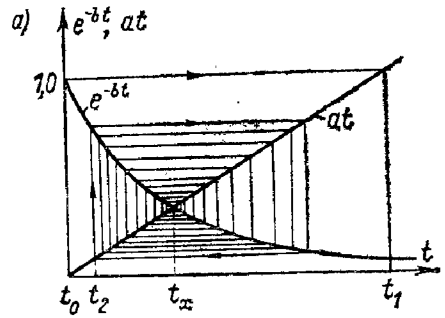


Рис. 1.7. Нахождение корня трансцендентного уравнения

ка обычно характеризует точность, достигнутую на очередном шаге вычислений. Поскольку используются сходящиеся процессы вычислений, то мерой точности может быть разность значений вычисляемой величины на настоящем и предыдущем циклах.

Рассмотрим в качестве примера алгоритм нахождения корня t_x трансцендентного уравнения $e^{-bt_x} = at_x$. Графики функций, входящих в уравнение, приведены на рис. 1.7, *а*. Корнем уравнения является абсцисса точки пересечения $t=t_x$. Зададим первоначальное приближение значения корня, например $t_x \approx t_0 = 0$. Тогда можно вычислить значение $M_0 = e^{-bt_0}$ и определить соответствующий аргумент правой части уравнения: $t_1 = M_0/a$. Эта величина принимается за второе приближение значения корня. Повторение вычислений с этим значением позволяет определить третье приближение $t=t_2$ и т. д. Из рис. 1,7, *а* ясно, что процесс является сходящимся и значение t_i может сколь угодно приближаться к t_x. Мерой точности является разность $\Delta t = t_{i+1} - t_i$ предыдущего и последующего результатов. Сравнивая ее с заданной мерой точности δ, можно принять решение о прекращении вычислений и выводе результата. Схема рассмотренного алгоритма приведена на рис. 1.7, *б*. Использованное в блоке обозначение $t_i:=t_{i+1}$ (оператор присваивания) означает, что повторение вычислений в цикле нужно вести с новым значением t_i, т. е. в регистр, где была записана величина t_i, следует записать новое значение t_{i+1}, для чего в программе должна быть предусмотрена специальная команда.

Основными критериями оценки алгоритма являются скорость сходимости итерационного процесса и требуемый объем вычислений. Эти требования во многом противоречивы. Малая скорость сходимости решений требует увеличения числа циклов для достижения требуемой точности, что увеличивает время вычислений. Ускорение сходимости требует, как правило, усложнения алгоритма, что, в свою очередь, увеличивает время выполнения каждого цикла и дополнительно ужесточает требования к емкости памяти ЭВМ. Поэтому оптимизация алгоритмов в общем виде едва ли возможна. В отношении рассмотренного примера отметим простоту алгоритма, но относительно большое число требуемых приближений.

Архитектура МК. После того как схема алгоритма разработана и по возможности оптимизирована, необходимо составить программу вычислений, пригодную для введения и выполнения в конкретной ЭВМ. В частности, в программе не должно быть операций, не имеющихся в составе системы команд машины. Алгоритм, записанный на понятном вычислительной машине языке, и называется программой. Перевод записи готового алгоритма на язык машины является формальным процессом, допускающим его автоматизацию. Однако для осуществления этого этапа необходимо иметь точное представление о возможностях общения оператора с машиной, что обычно выражается в понятии архитектуры ЭВМ, включающем в себя все то, что машина предоставляет программисту [22]. Наиболее важными элементами архитектуры ЭВМ являются система команд, их форматы, способы адресации, структура памяти, способы представления чисел и др. К архитектуре не относятся особенности технической реализации машины, ее быстродействие, эксплуатационные характеристики. Так, способ вычисления тригонометрических функций не представляет интереса для пользователя, для него важно наличие в МК клавиши, после нажатия которой последует вычисление, например, sin *x*.

Действие ЭВМ сводится к выполнению команд, задающих операции. Команда содержит в себе сведения о том, какая операция должна быть выполнена, над какими операндами и куда должен быть помещен результат. Поэтому команда включает в себя код операции (КОП) и адресную часть. Способы интерпретации сведений в адресной части называют способами адресации. Используемая система команд и способы адресации представляют собой язык ЭВМ, являющийся важнейшей составной частью ее архитектуры.

Виды команд МК. Вне зависимости от языка ЭВМ по функциональному признаку различают три вида команд. К о м а н д ы п е р е с ы л к и обеспечивают обмен данными между различными ячейками памяти в составе ЭВМ. Применительно к МК к этой группе относятся, например, команды записи содержимого РХ в регистры памяти.

К о м а н д ы у п р а в л е н и я осуществляют управление порядком прохождения программы в соответствии со значениями внешних воздействий на МК или с выработанными в процессе вычислений промежуточными результатами. К командам управления относится, например, команда «Стоп — Пуск».

Среди команд управления выделяют команды условного и безусловного переходов и перехода к подпрограмме. Общей чертой всех этих команд является необходимость указания адреса перехода, т. е. того показания программного счетчика, при котором после перехода будет выполняться следующая команда программы. Поэтому команды перехода в МК занимают два шага программы: собственно команда и адрес перехода. Условные переходы осуществляются при разветвляющемся алгоритме. Поскольку в МК результат предыдущих вычислений находится в РХ, то команды условных переходов осуществляют переход в зависимости от содержания этого регистра. Обычно предусматриваются переходы в зависимости от условий $x<0$, $x=0$, $x\geqslant0$, $x\neq0$. Используется инверсная логика перехода: если условие выполняется, переход не осуществляется, если условие не выполняется, то происходит переход в программе к команде по указанному адресу.

Безусловный переход также осуществляется по адресу, указанному после команды перехода. Необходимость команды безусловного перехода возникает, например, когда программа составляется из фрагментов, записанных в различных зонах программного ОЗУ. Тогда последней командой предыдущего фрагмента должна быть команда безусловного перехода на первую команду следующего фрагмента. Другим использованием команды безусловного перехода является возврат к началу программы, например, для повторения вычислений с новыми исходными данными. Отметим также возможность использования команды безусловного перехода при циклических алгоритмах. Так, в цикле с верхним окончанием (см. рис. 1.6, *а*) после команды, соответствующей блоку *3* (тело цикла) требуется команда безусловного перехода на блок *1*. В схеме (см. рис. 1.6, *б*) такой команды не нужно: переход осуществляется условно. Необходимость в безусловном переходе возникла бы, если после логического блока *3* до возврата в тело цикла *2* потребовалась бы какая-либо операция (блок *5* на рис. 1.6, *б*). Соответствующую команду потребовалось бы поместить в некоторую ячейку программного ОЗУ, и дальнейший возврат к телу цикла *2* потребовал бы команды безусловного перехода.

Команда перехода к подпрограмме отличается тем, что кроме адреса перехода, соответствующего первой команде подпрограммы, требуется запоминание адреса возврата, т. е. показания программного счетчика, с которого продолжится выполнение программы после завершения подпрограммы. Этот адрес запоминается автоматически в адресном стеке. Для возвращения в основную программу каждая подпрограмма оканчивается специальной командой «Возврат». Стековая организация ЗУ возврата из подпрограммы (см. рис. 1.2) обеспечивает многократное вложение подпрограмм друг в друга.

Третьим видом команд являются к о м а н д ы о б р а б о т к и, вызывающие выполнение операций из числа предусмотренных для данной ЭВМ.

Адресация команд. Информация в адресной части команд содержит указания о местоположения операндов, результата операции, способе перехода к следующей команде, а также о возможной модификации элементов команды после ее выполнения.

Команды ЭВМ классифицируются по числу включаемых в команду адресов на одноадресные, многоадресные и безадресные.

Безадресные команды по существу являются командами с подразумеваемой адресацией, заложенной в конструкции ЭВМ. Аналогичным образом переход к следующему шагу программы в МК означает увеличение содержания программного счетчика на единицу. В командном слове (или в специальной команде перехода) отражаются лишь случаи, когда этот порядок нарушается. Команды пересылки в МК являются одноадресными. Участие в этих командах регистра X подразумевается.

Классифицировать команды можно по способу адресации, под которым понимается способ интерпретации содержания адресной части команды. Наиболее естественным является прямой способ адресации, когда в команде указывается или подразумевается адрес операнда и (или) результата. При косвенной адресации в адресной части команды содержится указание на регистр памяти, в котором записан адрес операнда. Поскольку в общем случае в ЭВМ число регистров памяти невелико, малая длина адресной части команды оказывается достаточной для записи их номеров, а в регистре памяти возможно записать более длинный адрес операнда, расположенного в памяти большого объема.

Другие распространенные способы адресации основаны на вычислении адреса операнда по какому-либо правилу и на изменении команд или адреса после выполнения операции. Изменение команд после их выполнения является важнейшим средством осуществления групповых операций, также возможно изменение кода операции, адресной части команды, признаков способа адресации. При этом во многих случаях адрес увеличивается или уменьшается на единицу. Такие способы адресации называются соответственно автоинкрементными и автодекрементными. Их использование обеспечивает последовательную обработку массивов данных и сокращение объема разрабатываемых программ. Особенно эффективно применение этих методов при организации счетчиков циклов.

Применительно к МК понятие формата команды ограничивается числом шагов в программной памяти, которое эта команда занимает. В большинстве случаев это один шаг (одна ячейка памяти программного ОЗУ). Исключение составляют, как отмечалось, команды переходов. Число нажатий клавиш для формирования кода команды может быть различным, однако это не имеет значения при программировании решения задачи.

Использование в МК различных методов адресации обеспечивает эффективность программирования. В командах обработки данных используется подразумеваемая адресация: размещение операндов и результата известно заранее. Наличие операционного стека с определенным порядком перемещения информации в нем расширяет возможности при программировании вычислений. Прямая адресация используется в командах обращения к регистрам памяти: команды состоят из кода операции и номера регистра, в который записывается или из которого извлекается информация. Косвенная адресация предусматривает включение в код команды номера регистра, в котором содержится адрес операнда, являющийся в МК адресом другого регистра. Использование в МК косвенной адресации эффективно в сочетании с модификацией адреса (автоинкрементной и автодекрементной), что значительно расширяет возможности программиста, облегчая организацию циклов и групповые операции. Таким образом, МК предоставляет программисту те же возможности, что и современные ЭВМ других классов.

Общие методические указания по применению МК. Использование МК обеспечивает возможность комплексного решения задач, связанных с числовыми расчетами. Вне зависимости от содержания задачи можно выделить следующие этапы ее решения:

1. Содержательное описание задачи, т. е. ее точная словесная формулировка с использованием однозначных математических понятий и определений.
2. Формализация содержательного описания в виде, пригодном для числовых расчетов.
3. Составление программы вычислений на языке, соответствующем выбранному МК.
4. Введение программы вычислений в память МК (если дальнейшая работа предполагает использование программы, записанной в ОЗУ МК).
5. Проверка правильности составления и введения программы (отметим, что этот этап необходим вне зависимости от того, выполняются вычисления по программе вручную или автоматически).
6. Проведение вычислений.
7. Анализ результатов.

Решение задачи с использованием средств вычислительной техники предполагает законченность действий: от словесной постановки задачи до получения результата. Рассмотренный план решения в значительной степени универсален и пригоден при использовании средств вычислительной техники любого класса: от МК и микропроцессорных систем до универсальных ЭВМ. В этом смысле работа с программируемыми МК полезна, так как способствует усвоению общих принципов и приобретению навыков общения человека с ЭВМ.

1.4. Формы представления чисел в микрокалькуляторах

В ЭВМ различных классов используются естественная и показательная форма представления чисел.

Любое число может быть записано в естественной форме как $a_n a_{n-1} \ldots a_0,\ b_0 b_2 \ldots b_m$. В случае когда число разрядов, отводимых на целую и дробную части числа, строго определено, говорят о представлении числа с фиксированной запятой (точкой). При возможности записи числа с произвольным расположением запятой говорят о представлении его с плавающей запятой. Преобразование числа с произвольным расположением запятой в форму, стандартную для данной ЭВМ, называется нормализацией числа.

В микрокалькуляторах Б3-21 и Б3-34 для введения чисел в естественной форме используется восемь разрядов индикатора с плавающей запятой. Следовательно, в этой форме можно записать числа от $\pm 0{,}0000001 = 10^{-7}$ до $\pm 99999999 = 10^8 - 1$. Предусмотрена автоматическая блокировка ввода при попытке ввести лишний значащий разряд.

Во многих задачах диапазон чисел, представленных в естественной форме, оказывается недостаточным, и используется показательная форма представления чисел: $a_0,\ b_0 b_1 \ldots b_p 10^r$, где нормализованный множитель $a_0,\ b_0 b_1 \ldots b_p$ называют мантиссой числа, а показатель r — его порядком. В Б3-21 и Б3-34 используется нормализация с условием $1 \leqslant a_0 \leqslant 9$. Для мантиссы отводится восемь разрядов индикатора, для порядка — два И мантисса, и порядок вводятся со знаками.

Для введения порядка используется клавиша «ВП». При этом в М могут быть представлены числа в диапазоне от $\pm 10^{-99}$ д $\pm 9{,}9999999 \cdot 10^{+99}$, что значительно превышает диапазон предста ления чисел в естественной форме. Поскольку изменение порядка э вивалентно изменению положения запятой, показательную форму пре ставления чисел часто отождествляют с представлением с плава щей запятой. Обычно пользователь может вводить числа в МК произвольной форме, в том числе с ненормализованной мантиссо а в процессе вычислений осуществляется автоматический перев числа в естественную форму, если $1 \leqslant |x| \leqslant 99999999$, и в показател ную — в остальных случаях.

Некоторые МК имеют так называемые скрытые разряды. Они п являются тогда, когда разрядность внутренних регистров боль числа позиций в индикаторе, отображающем число. Иногда соде жание скрытых разрядов является доступным пользователю. Дл установления факта доступности скрытых разрядов можно вызвать РХ из ПЗУ число π, а затем отнять от него целую часть. Число н индикаторе сдвинется на одну позицию влево, а появление на дев той позиции новой цифры будет свидетельствовать о наличии и д ступности скрытого разряда. В Б3-34 доступных скрытых разрядо нет. В некоторых выпусках Б3-21 такой разряд есть, что связано применением индикаторов, в которых для записи десятичной зап той используется целая позиция. Это обстоятельство не ограничива возможностей данного МК по сравнению с аналогичными калькул торами других выпусков с той лишь разницей, что при записи числ содержащего десятичную запятую, восьмая значащая цифра буде записана в регистр X, но на индикаторе ее видно не будет. Это к сается и результата вычислений. Вывод на индикатор скрытого ра ряда осуществляется при вычитании единиц старшего разряда числа

Как было отмечено, форма представления чисел с плавающе запятой позволяет оперировать в рассматриваемых МК с числами модуль которых находится в интервале от $1 \cdot 10^{-99}$ до $10^{99} \cdot 9{,}999999$ Что произойдет, если попытаться ввести в МК числа, выходящие з пределы этого диапазона? В микрокалькуляторах Б3-21 и Б3-34 мож но записать число, меньшее по модулю чем $1 \cdot 10^{-99}$, например, $0{,}1 \times 10^{-99}$, но попытка ввода этого числа в регистр Y (нажатие кла виши «↑») вызовет обнуление индикатора. При попытке записат число, большее по модулю чем $10^{99} \cdot 9{,}9999999$ в Б3-34 на индикато ре появится сигнал ошибки. В Б3-21 сигнал ошибки появится пр вводе этого числа в регистр Y. Следует отметить, что на некоторы Б3-21 можно выполнять одноместные операции с числами, больш ми по модулю чем $10^{99} \cdot 9{,}9999999$ и меньшими по модулю чем $1 \cdot 10^{-9}$ если результат вычислений оказывается внутри допустимого инте вала.

УСТРОЙСТВО И СИСТЕМА КОМАНД МИКРОКАЛЬКУЛЯТОРОВ Б3-21 И Б3-34

2.1. Элементы ввода, индикации и хранения информации в микрокалькуляторах

Клавиатура и индикатор. Основными элементами архитектуры отечественных программируемых МК являются устройства ввода — вывода, емкость и структура памяти и используемая система команд.

В МК ввод исходных данных, подача команд, определяющих выполнение операций с этими данными, и управление работой осуществляется с помощью клавиатуры. Отображение вводимых операндов и результатов выполнения операций обеспечивается индикатором.

Поясняющие надписи на лицевой панели микрокалькуляторов Б3-21 и Б3-34 (рис. 2.1 и 2.2) располагаются непосредственно на клавишах, на корпусе МК над клавишей и иногда под нею. Соответственно одна клавиша может использоваться для формирования одной, двух или трех команд. Выполнение операции, обозначенной на клавише, обеспечивается при ее нажатии. Для выполнения операций, обозначенных на корпусе МК, требуется предварительное нажатие так называемой префиксной клавиши, которая обеспечивает переход к операции, обозначенной над или под клавишей. Совмещение функций клавиши расширяет число возможных операций при ограниченных габаритах МК.

Команду, обозначенную на клавише, будем в дальнейшем записывать в соответствии с ее изображением: +; —; ↑; а команду, выполняемую при нажатии двух или трех клавиш, — с указанием используемых префиксных клавиш: Pln; Fx^2 (Б3-21); Fsin; КНОП; КИП2 (Б3-34).

В Б3-21 и Б3-34 используются двенадцатиразрядные светодиодные индикаторы. Два режима работы МК — «Работа» и «Программирование» — обусловливают два режима работы индикатора. В режиме «Работа», который всегда устанавливается при включении, индикатор отражает содержимое регистра X(РХ). Размещение числа в разрядах индикатора иллюстрируется рис. 2.3. Число вводится в РХ при последовательном нажатии цифровых клавиш и клавиши десятичной запятой. Попытк ввода в РХ более восьми цифр приводит к блокировк ввода: дальнейшее нажатие цифровых клавиш не восприн нимается (рис. 2.3, *а*).

При записи чисел в показательной форме (см. § 1.4 порядок числа представляется двумя правыми разрядам

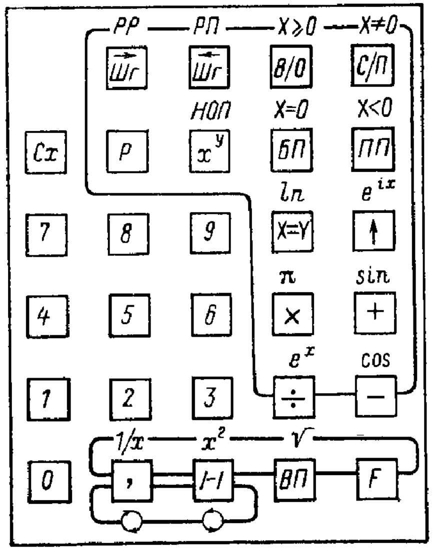


Рис. 2.1. Внешний вид клавиатуры Б3-21

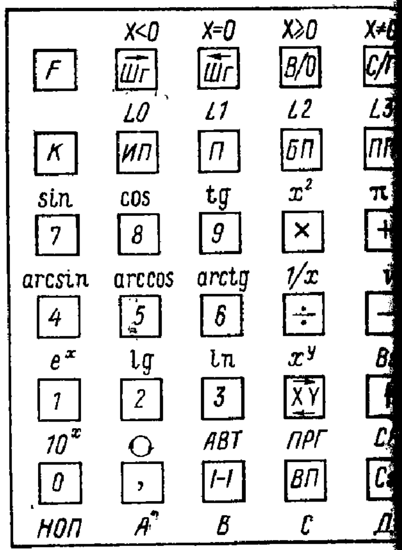


Рис. 2.2. Внешний вид клавиатуры Б3-34

индикатора (рис. 2.3, *в* — *е*). На индикаторе высвечива ется только отрицательный знак числа и порядка.

В первых выпусках Б3-21 использовался индикатор в котором отображение десятичной запятой занимае отдельный разряд (рис. 2.3, *ж*). В этом случае пользо ватель не видит восьмого разряда числа, хотя в РХ о существует. Для того чтобы вывести на индикатор это разряд, достаточно из записанного в РХ числа вычест значение старшего разряда. Число на индикаторе сдви нется на одну позицию влево и на девятой позиции по явится младший (скрытый) разряд.

Все вычисления в МК выполняются в режиме «Рабо та». Но для автоматического проведения вычислений не обходимо составленную программу записать в ОЗУ программ. Запись команд программы в ячейки памяти

ОЗУ выполняется в режиме «Программирование». В этом режиме индикатор МК представляет собой дисплей, отображающий содержимое некоторого участка (три ячейки памяти) ОЗУ (рис. 2.4).

Коды команд при их записи в ОЗУ появляются на индикаторе слева и при вводе очередной команды сдвигаются вправо. Две крайние правые позиции отражают

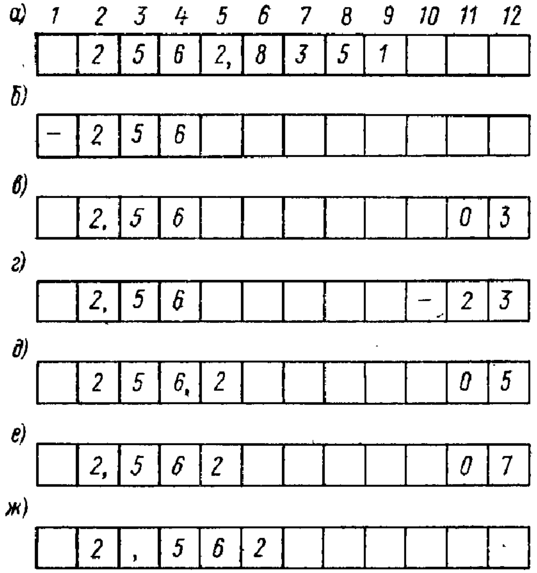


Рис. 2.3. Размещение числа в разрядах индикатора

содержание программного счетчика, т. е. номер той ячейки памяти, куда записывается следующая за наблюдаемыми команда. Исходное состояние программного счетчика 00 соответствует установке его на ячейку памяти с номером 00 (рис. 2.4, *а*). Рассмотрим пример вывода на индикатор некоторого фрагмента программы (рис. 2.4, *б*). Команда с кодом 12 записана в ячейку памяти с номером 50, с кодом 61 — в ячейку 51, с кодом 18 — в ячейку 52. Ввод в память очередных команд с кодами 86 и 06 вызывает сдвиг отображаемых кодов команд вправо и изменение состояния программного счетчика на единицу (рис. 2.4, *в*, *г*).

Структура памяти МК. В гл. 1 при рассмотрении структуры программируемых МК указывалось на наличие в них различных по исполнению, организации и функциональному назначению элементов памяти: постоянного запоминающего устройства (ПЗУ), оперативного запоминающего устройства (ОЗУ) для записи программ, операционных регистров, регистров памяти (РП), регистров стека. Количественные характеристики элементов памяти являются одними из важнейших характеристик

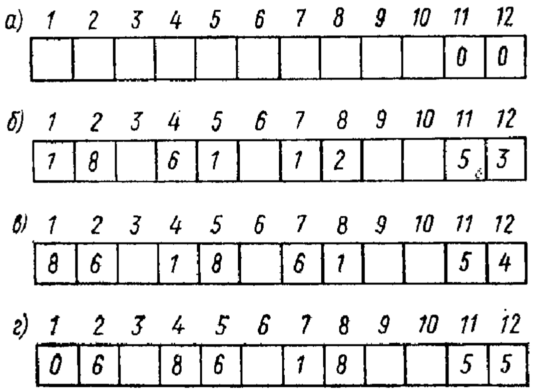


Рис. 2.4. Работа индикатора в режиме программирования

МК: перечень автоматически вычисляемых функций (ПЗУ), емкость памяти для записи программ (ОЗУ) и т. д. Особый интерес для пользователя представляет организация элементов оперативной памяти (ОЗУ, РП, стеки).

Память программы. Для записи команд программы используются ячейки памяти ОЗУ. В Б3-21 ОЗУ состоит из десяти «страниц», на каждой из которых имеется шесть ячеек памяти. Всего в ОЗУ Б3-21 можно записать программу, состоящую из 60 команд. Каждая ячейка памяти имеет свой двузначный номер, первая цифра которого означает номер «страницы» (от 0 до 9), а вторая — номер ячейки на «странице» (от 0 до 5). Таким образом, последняя ячейка ОЗУ в Б3-21 имеет номер 95, тогда как ее порядковый номер 60. В Б3-34 ОЗУ содержит 98 ячеек памяти с естественной нумерацией от 0 до 97.

Регистры памяти. Для запоминания исходных данных и промежуточных результатов вычисления ис-

пользуются регистры памяти. В микрокалькуляторе Б3-21 таких регистров 7 с номерами от 2 до 8, в Б3-34 — 14 с номерами от 0 до 9 и А, В, С, Д. Входной язык МК содержит команды, обеспечивающие как запись числа из РХ в РП, так и считывание его из РП в РХ.

Кольцевой стек. В структуре Б3-21 имеется шесть регистров, объединенных последовательно в кольцо через РХ (рис. 2.5). Подача соответствующих команд обеспечивает перезапись чисел из регистра в регистр против часовой стрелки (содержимое РХ переписывается в С6, содержимое С6 — в С5,.., С1 — в РХ) либо по часовой стрелке. Прямой доступ к регистрам С1—С6 не предусмотрен. Таким образом, чтобы вызвать в РХ содержимое, например, регистра С2, можно два раза подавать команду перезаписи числа в стеке против часовой стрелки либо пять раз команду перезаписи по часовой стрелке.

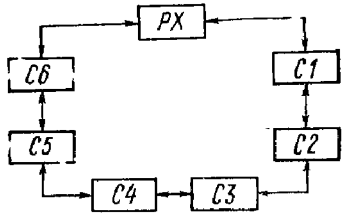


Рис. 2.5. Кольцевой стек Б3-21

Регистры стека при выполнении вычислений могут использоваться так же, как РП. Подача команд кольцевого сдвига обеспечит вызов содержимого регистров С1 — С6 в РХ. При этом, чтобы избежать ошибок, необходимо четко знать местоположение того или иного числа в стеке и его перемещение там в процессе вычислений. При работе с кольцевым стеком в отличие от работы с РП вызов некоторого числа из стека в РХ ведет к изменению информации в соответствующем регистре стека. Вместе с тем ранее записанное в РХ число не теряется, а переносится в стек. Стек удобно использовать для хранения данных, которые необходимо иметь в памяти МК только до их первого считывания. При вызове в РХ информации, когда записанное там число не должно быть потеряно, с помощью стека одна команда реализует сразу два действия: содержимое С1 или С6 переходит в РХ, а число из РХ заносится в С6 или С1. Использование же РП потребовало бы подачи в этом случае двух команд.

Использование стековой памяти в первых выпусках Б3-21 осложнено тем обстоятельством, что при вычислении $\sin x$, $\cos x$, $\ln x$, e^x, e^{ix}, x^y регистры С1 — С6 исполь-

Таблица 2.1

Аргумент функций $\sin x$ $\cos x$	n	Аргумент функции e^x	n	Аргумент функции e^{ix}	n
$0-\pi$	6	$0-3\ln 10$	—	$0-\pi$	—
$\pi-3\pi$	2	$3\ln 10-9\ln 10$	2	$\pi-3\pi$	4
$3\pi-10$	—	$9\ln 10-10\ln 10$	—	$3\pi-10$	—
$10-5\pi$	3	$10\ln 10-11\ln 10$	6	$10-5\pi$	5
$5\pi-7\pi$	—	$11\ln 10-12\ln 10$	1	$5\pi-7\pi$	1
$7\pi-9\pi$	5	$12\ln 10-13\ln 10$	6	$7\pi-9\pi$	—
$9\pi-10\pi$	1	$13\ln 10-14\ln 10$	1	$9\pi-10\pi$	3
$10\pi-11\pi$	4	$14\ln 10-15\ln 10$	6	$10\pi-11\pi$	6
$11\pi-13\pi$	—	$15\ln 10-16\ln 10$	1	$11\pi-13\pi$	2
$13\pi-16\pi$	6	$16\ln 10-17\ln 10$	6	$13\pi-15\pi$	—

зуются как рабочие и ранее записанные в них числа могут быть потеряны. Момент занесения в эти регистры служебной информации зависит от вычисляемой функции и значения аргумента. Сведения, приведенные в табл. 2.1 [28], позволяют в некоторых случаях сохранить возможность использования стека при выполнении программ, содержащих вычисление функций $\sin x$, $\cos x$, e^x, e^{ix}. В табл. 2.1 в графе, обозначенной буквой n, записан номер регистра стека, содержимое которого в процессе вычисления указанных функций будет утрачено.

Операционные регистры. Взаимодействие между операционными регистрами организовано по принципу стека. Можно объединить эти регистры понятием операционного стека. Во входном языке МК имеются команды, обеспечивающие запись числа из РХ в РY и обмен числами в этих регистрах. При вычислениях оказывается важным то обстоятельство, что перемещение информации из РХ в РY в некоторых ситуациях выполняется автоматически. В Б3-34 в операционный стек кроме регистров РХ и РY входят регистры PZ и РТ, а также регистр предыдущего результата РХ1.

Необходимым условием исключения ошибок при программировании является знание правил перемещения информации в регистрах операционного стека (табл. 2.2, 2.3). Одновременно знание этих правил позволяет упростить программы вычислений, исключая из них избыточ-

Таблица 2.2

Выполняемая операция	Состояние регистров после выполнения операции	
	РХ	РУ
Исходное состояние	A	B
Команда «Стирание» Сx	0	B
Команда «Ввод» ↑	A	A
Обмен информации XY	B	A
Набор числа C после команды Сx	C	B
Набор числа C после операций обработки	C	A
Операция обработки, кроме e^{ix}	Результат операции	B
Вычисление e^{ix}	cos A	sin A
Управление	A	B
Пересылка из РХ в РП	A	B
Пересылки в РХ числа C из РП или π из ПЗУ	C (π)	B

ные операции пересылки, а также сокращая число занятых РП. Примеры использования регистров операционного стека при вычислениях рассмотрены в гл. 4.

Таблица 2.3

Выполняемая операция	Состояние регистров после выполнения операции				
	РХ 1	РХ	РY	PZ	РТ
Исходное состояние	A	B	C	D	I
Команда «Стирание» Сx	A	0	C	D	I
Команда «Ввод» ↑	A	B	B	C	D
Обмен информации XY	A	C	B	D	I
Кольцевая перезапись $F \to$	B	C	D	I	B
Одноместная операция и x^y	B	результат операции	C	D	I
Двухместная операция, кроме x^y	B	результат операции	D	I	I
Запись числа N в РХ из ПЗУ или РП	A	N	B	C	D
Запись числа из РХ в РП	A	B	C	D	I
Вызов предыдущего результата FBx	A	A	B	C	D
Набор числа N после операции Сx и ↑	A	N	C	D	I

Стек возврата. В Б3-21 и Б3-34 имеется стек, который недоступен пользователю. Его назначение — обеспечение возможности программирования с использованием подпрограмм. Этот стек с магазинной организацией на пять регистров. По команде перехода к подпрограмме в нем записывается адрес продолжения основной программы — адрес возврата из подпрограммы. Работа с подпрограммами в программируемом режиме подробно рассмотрена ниже.

2.2. Система команд микрокалькуляторов Б3-21 и Б3-34

Команды пересылки. При вычислениях между регистрами МК осуществляется обмен информацией. Это происходит автоматически либо по командам, вводимым с клавиатуры. Команды, обеспечивающие обмен информацией между регистрами, называются командами пересылки и представляют собой один из трех классов команд входного языка вычислительного средства. В табл. 2.4 представлены команды пересылки Б3-21 и Б3-34.

Ввод числа, записанного в естественной форме в регистр РХ, осуществляется последовательным нажатием цифровых клавиш и клавиши десятичной запятой (табл. 2.4, пп. 1, 2). Постоянная π часто используется в различных расчетах, поэтому в МК предусмотрено хранение этого числа в постоянном запоминающем устройстве (ПЗУ) и возможен вызов его в регистр РХ с помощью специальной команды (табл. 2.4, п. 3).

Для стирания содержимого РХ используется команда Сx (табл. 2.4, п. 9). При этом в РХ записывается ноль, который высвечивается только во втором разряде индикатора.

Перемещение информации в регистрах РХ и РY обеспечивается командами ↑ и ХY (табл. 2.4, пп. 4 и 5). В Б3-34 команда ↑ выполняет перемещение чисел во всех четырех регистрах операционного стека (см. табл. 2.3).

Выполнение команд копирования (табл. 2.4, пп. 6, 7 и 10) не изменяет содержимого регистра, из которого копирование производится. В Б3-21 регистрам X и Y присвоены номера 0 и 1 соответственно, что и объясняет нумерацию РП начиная с двух. Формально существуют команды Р0, Р1, F0 и F1. Команды Р0 и F0 реализуют обмен содержимого РХ с самим собой. Команда Р1 эквивалентна команде ↑. Команда F1 отличается от команды XY тем, что при ее выполнении содержимое РХ теряется.

В Б3-34 имеется регистр предыдущего результата РХ1. Перемещение числа в этот регистр осуществляется автоматически (см. табл. 2.3). Вызов числа из РХ1 в РХ выполняется командой FВx (табл. 2.4, п. 10).

После команд пересылки (табл. 2.4, пп. 3—8, 10) запись нового числа в РХ не требует предварительного стирания его содержимого, так как оно выполняется автоматически.

Команды пересылки позволяют занести в регистры МК числа, над которыми в ходе вычислений выполняются некоторые действия — операции обработки. Участвующие в очередной операции числа долж-

Таблица 2.4

Номер пп.	Используемые обозначения команд для микрокалькуляторов		Выполняемая операция
	Б3-21	Б3-34	
1	0, ..., 9	0,...,9	Вызов из ПЗУ в РХ цифр 0—9
2	,	,	Вызов из ПЗУ в РХ знака десятичной запятой
3	Рπ	Fπ	Вызов из ПЗУ в РХ числа π
4	↑	↑	Ввод числа из РХ в РY, или разделение чисел
5	XY	XY	Обмен информацией между регистрами X и Y
6	Р2, ..., Р8	П0,..., ПД	Копирование числа, находящегося в РХ, в РП, номер которого указан в команде
7	F2,...,F8	ИП0,...,ИПД	Копирование числа из РП, номер которого указан в команде, в РХ
8	Р→ Р←	F→	Кольцевая перезапись чисел в регистрах стека по ходу часовой стрелки и против
9	Сx	Сx	Стирание содержимого РХ с записью в него нуля
10	—	FВx	Копирование в РХ содержимого регистра предыдущего результата Х1

ны быть записаны в РХ при одноместных операциях или в РХ и РY при двухместной операции.

Команды обработки. Для выполнения над числами, записанными в операционные регистры, математических действий, перечень которых определен входным языком МК, используются команды обработки. Эти команды для Б3-21 и Б3-34 сведены в табл. 2.5. Результат вычислений всегда помещается в регистре X. Исключение составляет операция вычисления функции e^{ix} в Б3-21. Функция e^{ix} представляется в виде $\cos x + i \sin x$. Значение $\cos x$ записывается в РХ, значение $\sin x$ — в РY (см. табл. 2.2).

Значения аргументов тригонометрических функций в Б3-21 представляются в радианной мере. В Б3-34 значения аргументов тригонометрических функций и результаты вычисления обратных тригонометрических функций представляются в радианах или градусах в зависимости от положения переключателя «Р—Г» на лицевой панели МК в момент подачи команды на вычисление соответствующих функций.

При выполнении двухместной операции два операнда размещаются в регистрах X и Y. При выполнении операции сложения и умножения последовательность ввода чисел в МК безразлична. При

Таблица 2.5

Номер пп.	Используемые обозначения команд для микрокалькуляторов		Выполняемая операция
	Б3-21	Б3-34	
1	$F\frac{1}{x}$	$F\frac{1}{x}$	Нахождение обратной величины числа, записанного в РХ
2	Fx^2	Fx^2	Возведение числа в квадрат
3	$F\sqrt{\ }$	$F\sqrt{\ }$	Извлечение из числа квадратного корня
4	Fsin	Fsin	Вычисление функции sin x
5	Fcos	Fcos	Вычисление функции cos x
6	—	Ftg	Вычисление функции tg x
7	—	Farcsin	Вычисление функции arcsin x
8	—	Farccos	Вычисление функции arccos x
9	—	Farctg	Вычисление функции arctg x
10	Fln	Fln	Вычисление натурального логарифма числа
11	—	Flg	Вычисление десятичного логарифма числа
12	Fe^x	Fe^x	Вычисление функции e^x
13	—	$F10^x$	Вычисление функции 10^x
14	Fe^{ix}	—	Вычисление функции e^{ix}
15	/—/	/—/	Перемена знака числа
16	+	+	Сложение
17	−	−	Вычитание
18	×	×	Умножение
19	÷	÷	Деление
20	x^y	x^y	Вычисление функции x^y

выполнении остальных операций в РY должны находиться уменьшаемое, делимое и показатель степени. Операция выполняется по алгоритму: запись в РХ операнда y, пересылка операнда в РY командой ↑, запись в РХ операнда x, команда на выполнение операции.

После выполнения операций обработки в Б3-21 ввод в РХ нового числа сопровождается перемещением ранее записанного в нем числа в РY без дополнительной команды ↑. В Б3-34 такое перемещение выполняется и после команд пересылки. Это существенно облегчает выполнение нескольких операций подряд.

Использование команд пересылки и обработки, а также подготовительной команды для ввода порядка числа (ВП), относящейся к классу команд управления, дает возможность вручную осуществлять выполнение любой программы, доступной данному МК.

Команды управления. Программа, предназначенная для записи в ОЗУ, по сути не отличается от программы вычислений, реализуемой вручную. Однако организация ее записи и выполнения требует применения нового класса команд — команд управления. Перечень команд управления Б3-21 и Б3-34 приведен в табл. 2.6. Эти команды можно разделить на три группы: команды управления МК (команды 1—4); команды управления программой (команды 9—30); коман-

Таблица 2.6

Номер пп.	Используемые обозначения команд для микрокалькуляторов		Выполняемая операция
	Б3-21	Б3-34	
1	РРП	FПРГ	Перевод МК в режим «Программирование»
2	РРР	FАВТ	Перевод МК в режим «Работа»
3	$\overrightarrow{ШГ}$ $\overleftarrow{ШГ}$	$\overrightarrow{ШГ}$ $\overleftarrow{ШГ}$	Пошаговый просмотр программы на индикаторе в сторону увеличения или уменьшения номеров состояний программного счетчика
4	ПП	ПП	Режим «Работа»: пошаговое прохождение программы
5	ВП	ВП	Предварительная команда введения порядка числа
6	С/П	С/П	Команда пуска и останова автоматического выполнения программы
7	В/О	В/О	А. Режим «Работа»: установ программного счетчика в нулевое состояние. Б. Режим «Программирование»: возврат из подпрограммы. В. Режим «Программирование», команда В/О вне подпрограммы: переход по адресу 01
8	БП, М	БП, М	Безусловный переход на адрес М
9	ПП, М	ПП, М	Режим «Программирование»: переход к выполнению подпрограммы, записанной, начиная с адреса М

Номер пп.	Используемые обозначения команд для микрокалькуляторов		Выполняемая операция
	Б3-21	Б3-34	
10 11 12 13	РX⩾0, М РX=0, М РX≠0, М РX<0, М	FX⩾0, М FX=0, М FX≠0, М FX<0, М	Условный переход по условию, указанному в команде; М — адрес перехода в случае невыполнения условия
14	РНОП	КНОП	Команда «Нет операции»
15	—	FL0, M—FL3, M	Организация цикла с регистрами Р0—Р3; М — адрес перехода
16	—	КБП0—КБПД	Косвенный безусловный переход по модифицированному адресу, указанному в регистре 0 — Д
17 18 19 20	—	КX⩾00— —КX⩾0Д КX=00— —КX=0Д КX≠00— —КX≠0Д КX<00— —КX<0Д	Косвенный условный переход по модифицированному адресу, указанному в регистре 0 — Д
21	—	КПП0—КППД	Косвенный переход к подпрограмме по модифицированному адресу, указанному в регистре 0—Д
22	—	КП0—КПД	Косвенная запись содержимого РX в РП по модифицированному адресу, указанному в регистре 0—Д
23	—	КП↑	Команда эквивалентна команде КП0 без модификации адреса
24	—	КИП0—КИПД	Косвенный вызов в РX содержимого РП, модифицированный адрес которого указан в регистре 0—Д

Номер пп.	Используемые обозначения команд для микрокалькуляторов		Выполняемая операция
	Б3-21	Б3-34	
25	—	КИП↑	Команда эквивалентна команде КИП0 без модификации адреса
26	—	КПП↑	Эквивалентна команде КПП0 без модификации адреса перехода
27 28 29 30	—	КX⩾0↑ КX=0↑ КX≠0↑ КX<0↑	Команды эквивалентны командам КX⩾00 КX=00 КX≠00 КX<00 без модификации адреса перехода

ды смешанного типа (команды 5—8). Рассмотрим назначение команд управления.

Первая и вторая команды табл. 2.6. предназначены для установления в МК одного из двух возможных режимов: «Работа» или «Программирование». При включении МК, как уже указывалось, всегда устанавливается режим «Работа». В этом режиме производятся и все вычисления. Запись программы вычислений в память программ осуществляется в режиме «Программирование».

В ходе работы может возникнуть необходимость просмотра записанной в ОЗУ программы либо ее части. Для этого в режиме «Программирование» используются команды $\overrightarrow{\text{ШГ}}$ и $\overleftarrow{\text{ШГ}}$ (п. 3), которые увеличивают или уменьшают число в программном счетчике на единицу.

Команда ВП (п. 5) обеспечивает ввод показателя степени числа.

Команда ПП (п. 4) предназначена для пошагового прохождения программы в режиме «Работа». Каждая подача этой команды инициирует выполнение одного очередного шага программы, записанной в ОЗУ.

Команда С/П (п. 6) обеспечивает пуск и останов вычислений по программе, записанной в ОЗУ МК. Эта команда должна обязательно завершать любую записываемую в ОЗУ программу для фиксации полученного результата. В режиме «Работа» она обеспечивает запуск автоматических вычислений, а также останов вычислений в случае их «зацикливания» — выхода на бесконечно повторяющийся цикл вычислений.

Команда В/О (п. 7) предназначена для обнуления программного счетчика (режим «Работа»), что позволяет выйти на начало записанной программы либо подготовиться к записи новой программы. В общем случае программа в память МК может быть записана в

ОЗУ с ненулевого адреса М, тогда поиск нужной ячейки памяти будет выполнен командой БП, М.

Когда команда В/О включена в программу, записанную в ОЗУ, ее действие может быть двояким. Во-первых, этой командой должна завершаться любая подпрограмма. Она обеспечивает обращение к стеку возврата, в котором записан адрес прерывания основной программы. Во-вторых, команда В/О, записанная в ОЗУ вне подпрограммы, обусловливает переход по адресу 01.

Команда РНОП или КНОП (п. 14) называется командой «Нет операции». Она может включаться в программу при отладке последней.

В табл. 2.6 присутствуют команды, состоящие из двух элементов: собственно команды и некоторого адреса М (пп. 8—13, 15). Адрес М называется адресом перехода. Наличие его в структуре указанных команд обязательно.

Команда безусловного перехода БП, М (п. 8) используется как при составлении программы для автоматических вычислений, так и в режиме «Работа» для поиска нужного места записанной в ОЗУ программы. При выполнении этой команды программный счетчик устанавливается в состояние М.

Команда перехода к подпрограмме ПП, М (п. 9) обеспечивает переход к первой команде подпрограммы с автоматическим запоминанием адреса возврата.

При решении задач может оказаться, что на определенном этапе возникает два различных по содержанию пути достижения результата. Выбор того или иного пути осуществляется исходя из некоторого условия. В зависимости от результата проверки этого условия происходит скачкообразное изменение состояния программного счетчика, обеспечивающее переход к нужному участку программы в сторону увеличения или уменьшения номера адресов. Автоматическая реализация таких программ в МК обеспечивается наличием команд условных переходов по условиям $X \geqslant 0$, $X < 0$, $X = 0$, $X \neq 0$. Очевидно, любые условия задачи при помощи алгебраических преобразований с соответствующими константами можно привести к четырем перечисленным выше. При выполнении указанного в команде условия число в программном счетчике увеличивается на два. Если условие не выполняется, то в программном счетчике устанавливается адрес М, записанный после команды проверки условия.

Команды, помещенные в табл. 2.6 в п. 15, имеют место только в Б3-34. Это команды организации циклов с регистрами Р0—Р3. При подаче команды FL0, М выполняются следующие операции: содержимое регистра Р0 уменьшается на единицу; новое содержимое Р0 сравнивается с нулем; если содержимое Р0 не равно 0, происходит переход в программе по адресу М; если содержимое Р0 равно нулю, выполняется команда, стоящая вслед за адресом М. Команду FL0, М можно заменить последовательностью нескольких рассмотренных команд: FL0, М≡ИП0; 1; —; П0; FX=0, М, где М — адрес перехода. При применении команд организации циклов происходит существенная экономия памяти.

Особенностью команд организации циклов является то, что при их выполнении не происходит изменения информации в операционных регистрах (замена команды описывающим ее фрагментом сопровождается изменением содержимого регистров операционного стека).

Рассмотренные команды условного и безусловного переходов, перехода к подпрограмме, организации циклов включают некоторый

адрес М, который записывается в свободную ячейку ОЗУ. Эти команды оказываются составными и занимают две ячейки в памяти программ. Способ адресации, используемый в названных командах, является прямым. В Б3-21 предусмотрена специфическая система адресации: в качестве адресов используются команды входного языка, например, XY, либо не имеющие иного смысла сочетания из двух нажимаемых клавиш, например F↑. В результате в ячейку памяти после смысловой части команды записывается некоторый код, соответствующий адресу перехода. Этот код на единицу превышает требуемый адрес; так, переход на нулевую ячейку с адресом 00 осуществляется набором адреса с помощью клавиш «Р» и «0», что дает на индикаторе код 01.

Команды, которые могут использоваться в качестве адресов перехода в Б3-21, приведены в табл. 2.7. В Б3-34 адрес набирается с помощью двух цифровых клавиш, например: 0,5 — переход по адресу 05; 3,8 — переход по адресу 38.

Следует отметить, что повторный ввод адреса М составной команды требует ввода и ее смысловой части.

В Б3-34 имеются команды, соответствующие командам условного и безусловного переходов, перехода к подпрограмме, а также обращения к регистрам памяти, но с иным видом адресации — косвенной адресацией (табл. 2.6, пп. 16—30). Наличие еще одного способа адресации в Б3-34 является несомненным достоинством этого МК.

Косвенная адресация предполагает предварительное занесение адреса перехода в один из регистров МК. Выполнение команды косвенной адресации представляет собой обращение к соответствующему регистру памяти, извлечение из него адреса перехода и изменение состояния программного счетчика в соответствии с этим адресом. Команды косвенной адресации могут быть заменены соответствующими командами прямой адресации. Например: команда БП, М эквивалентна команде КБП8, где адрес М записан в Р8. Таким образом, команда косвенной адресации, выполняя те же функции в про-

Таблица 2.7

Адрес перехода	Команда	Адрес перехода	Команда	Адрес перехода	Команда	Адрес перехода	Команда	Адрес перехода	Команда
00	Р0	20	Р2	40	Р4	60	Р6	80	Р8
01	F0	21	F2	41	F4	61	F6	81	F8
02	Р↑	22	Р×	42	Р,	62	РВП	82	Р—
03	0	23	2	43	4	63	6	83	8
04	F↑	24	F×	44	F,	64	FВП	84	F—
05	↑	25	×	45	,	65	ВП	85	—
10	Р1	30	Р3	50	Р5	70	Р7	90	Р9
11	F1	31	F3	51	F5	71	F7	91	F9
12	РXY	32	Р÷	52	Р/—/	72	РCx	92	Р+
13	1	33	3	53	5	73	7	93	9
14	FXY	34	F÷	54	F/—/	74	FCx	94	F+
15	XY	35	÷	55	/—/	75	Cx	95	+

грамме, что и команда прямой адресации, занимает одну ячейку памяти вместо двух.

Дополнительные возможности применения команд косвенной адресации связаны с модификацией адреса команды. Приведенное выше описание работы МК при выполнении команд косвенной адресации имеет место, когда адрес перехода или номер адресуемого регистра записан в Р7—РД. Использование РП с меньшими номерами связано с модификацией записанного в них адреса. В режиме косвенной адресации при обращении к регистрам Р0—Р3 их содержимое уменьшается на единицу; при обращении к регистрам Р4—Р6 их содержимое увеличивается на единицу. Таким образом, при каждом новом обращении выполняется переход по адресу, записанному в регистре с номером на единицу меньшим или большим предыдущего. Ячейки памяти, которые при обращении к ним автоматически увеличивают или уменьшают на единицу записанное в них число, а равно операции автоматического увеличения или уменьшения содержимого ячейки памяти на единицу, носят название соответственно автоинкрементных и автодекрементных.

Косвенное обращение через регистр Р0 как с модификацией, так и без модификации его содержимого, когда вместо команды, например, КП0 подается команда КП↑ (см. табл. 2.6, пп. 23, 25—30) расширяет возможности программирования на МК.

2.3. Исправление ошибок и некорректные операции

Если при наборе числа сделана ошибка, необходимо командой Сx стереть содержимое РХ и ввести число снова. Если ошибка возникла при вводе порядка числа, то необходимо еще раз набрать двузначный код требуемого порядка «*кр*», где «*к*» для однозначного показателя степени равно 0 (см. § 2.1). Если ошибочно нажата клавиша «/—/» — перемена знака числа, то ее необходимо нажать повторно.

При ошибочно нажатой префиксной клавише «F» в БЗ-34 необходимо подать команду CF.

В БЗ-21 действие ошибочно нажатой клавиши Р снимается нажатием клавиши «$\overrightarrow{\text{ШГ}}$». При ошибочно нажатой клавише «F» надо подать команду РРР. Если клавиши «F» и «Р» нажаты одна вместо другой, то нажимается нужная клавиша и продолжаются вычисления.

В общем случае пользователем может быть поставлена такая задача, которая при определенных исходных данных не имеет решения, например деление на нуль, либо задача, которую МК «не умеет» решать, например, извлечение корня квадратного из отрицательного числа. Такие операции называются некорректными.

К некорректным операциям относятся в БЗ-21 операции $\ln x$ при $x \leqslant 0$; $\sqrt{x}$ при $x<0$; x^y при $x \leqslant 0$ и деление на нуль. В БЗ-34, кроме перечисленных для БЗ-21, операции $\ln x$ при $x \leqslant 0$, $\operatorname{tg} x$ при $x=\pi/2 \pm n\pi$; $\arcsin x$ и $\arccos x$ при $|x|>1$. При задании некорректной операции на индикаторе в БЗ-21 появляется цифра «0» в первом (левом) разряде; а в БЗ-34 — слово ЕГГОГ. В этом случае необходимо подать команду Сx, после чего можно продолжать вычисления.

К некорректным операциям относится ввод в РХ числа, большего чем $10^{99} \cdot 9{,}9\,999\,999$. В БЗ-34 индикация некорректной операции возникает при наборе этого числа в РХ, а в БЗ-21 — при вводе его в РY. Последнее обстоятельство позволяет в БЗ-21 выполнить одноместные операции для значений x, больших, чем $10^{99} \cdot 9{,}9\,999\,999$, если результат операции меньше этого числа.

Индикация некорректной операции может свидетельствовать о том, что результат вычислений оказывается за пределами диапазона представления числа в МК, т. е. по модулю больше числа $10^{99} \times \times 9{,}9\,999\,999$. Если результат вычислений по модулю оказывается меньше, чем 10^{-99}, то на индикаторе высвечивается цифра «0». Это необходимо иметь в виду при выполнении продолжительных вычислений: нулевой промежуточный результат может привести к значительной ошибке в конечном результате (см. гл. 4).

Исключение из правила в БЗ-34 составляют операции вычисления e^x при $x \leqslant -231$ и 10^x при $x \leqslant -100$, когда вместо появления нуля регистрируется наличие ошибки.

Глава третья

ПРИЕМЫ ПРОГРАММИРОВАНИЯ

3.1. Составление простейшей программы

Общие положения. Составление программы является необходимым условием реализации вычислительного алгоритма в любом режиме — ручном или автоматическом. Запись разработанной программы вычислений в ОЗУ имеет смысл при многократном ее применении, например, при вычислении значений функции для различных значений ее аргумента с целью построения графика. Кроме экономии труда и времени, автоматическое повторение вычислений исключает неизбежные ошибки пользователя.

При составлении сложной программы и ее отладке целесообразно использовать подробную запись, форма которой представлена табл. 3.1. В этой таблице во второй графе записываются адреса ячеек памяти, в которых располагаются команды программы. При организации разветвляющихся и циклических программ вторая графа позволяет определить требуемое изменение содержания программного счетчика. В третьей графе записываются команды программы, а в четвертой графе — коды

Таблица 3.1

Номер пп.	Адрес	Команда	КОП	Содержимое регистров в операционном стеке	РП	Комментарий

выполняемых операций (КОП). Эта графа дает возможность контролировать правильность ввода программы в ОЗУ путем наблюдения последовательности КОП на индикаторе в режиме «Программирование», и вводить ее целесообразно после окончания программирования.

В пятую графу вносится содержимое регистров операционного стека (РХ и PY для Б3-21 и РХ1, РХ, PY, PZ, РТ — для Б3-34), которое необходимо знать при отладке сложных программ. В тех случаях, когда содержимое отдельных регистров операционного стека в программе никак не используется, соответствующие позиции таблицы можно не заполнять.

В шестую графу записываются регистры памяти (РП) в момент записи в них исходных данных или промежуточных результатов на определенном этапе вычислений, что позволяет ориентироваться в размещении этих данных и в имеющемся резерве свободных РП.

В графе 7 поясняется действие отдельных команд или фрагментов программы. Излишней подробности комментариев следует избегать.

Процесс программирования обычно проходит в несколько этапов с последовательным улучшением составляемой программы. Число этих этапов и совершенство окончательного варианта программы зависят от искусства программиста. Первоначальный вариант программы должен по крайней мере обеспечить выполнение условий задачи при выбранном алгоритме решения без учета требований экономии времени, числа используемых РП и шагов программы. Некоторые возможности улучшения программы выявляются еще до завершения программирования и могут быть учтены сразу.

Тщательная отработка программы целесообразна при многократном ее использовании. При этом достигается уменьшение времени вычислений или возможность использования освобождающегося резерва для усложнения решаемой задачи. Некоторые методы оптимизации программ рассматриваются в § 4.1.

Особое внимание при программировании должно быть уделено удобству пользования программой, которое определяется наличием в программе вспомогательных операций, сводящих к минимуму действия пользователя. Возможность введения в программу таких операций требует дополнительных ячеек памяти, поэтому усовершенствование программы в этом смысле выполняется в

последнюю очередь. Минимизация числа используемых РП и сокращение длины программы позволяет совершенствовать в дальнейшем вспомогательное обеспечение.

При программировании может оказаться, что программа все-таки превышает емкость имеющейся памяти. Если это превышение незначительно, то можно предпринять поиск более короткого пути достижения результата. В других случаях возможно разделить программу на части и использовать их комбинированно (см. § 4.1).

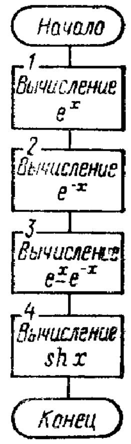


Рис. 3.1. Алгоритм вычисления sh x

Пример составления программы. Рассмотрим порядок составления программы на примере вычисления функции гиперболического синуса: sh $x = (e^x - e^{-x})/2$, алгоритм вычислений которой представлен на рис. 3.1. Дополнительно необходимо определить область изменения аргумента, в которой должно рассчитываться значение sh x. На основании алгоритма (рис. 3.1) может быть составлена программа для Б3-21, показанная в табл. 3.2. Проследим ход рассуждений при составлении этой программы.

Обычно принимают, что значение аргумента x перед началом выполнения программы должно быть занесено в РХ (исходное состояние). Поскольку при выполнении первой же операции e^x значение x будет потеряно, запишем его в РП (команда P2). Теперь в соответствии с алгоритмом (рис. 3.1) вычислим значение e^x (команда Pe^x). Величина e^x также может в дальнейшем понадобиться, поэтому поместим ее в память (команда P3).

Далее необходимо вычислить величину e^{-x}. С этой целью можно вызвать из P2 значение x, поменять его знак (команда /—/) и снова подать команду Pe^{-x}. Более коротким оказывается путь нахождения значения e^{-x} как обратной величины уже определенного нами значения e^x. При таком решении команда P2 для запоминания значения x оказывается избыточной.

На четвертом шаге программы необходимо определить разность $e^x - e^{-x}$. Для этого можно запомнить значение e^{-x} в РП, вызвать в РХ значение e^x, перевести его в PY, вызвать в РХ значение e^{-x} и выполнить вычитание.

Таблица 3.2

Адрес	Команда	РХ	РY	РП	Комментарий
		x			Исходное состояние
00	Р2	x		$x=$<Р2>	Запись значения x в Р2
01	$\text{Р}e^x$	e^x			Вычисление e^x
02	Р3	e^x		$e^x=$<Р3>	Запись значения e^x в Р3
03	$\text{F}1/x$	e^{-x}			Вычисление e^{-x}
04	↑	e^{-x}	e^{-x}		Ввод e^{-x} в РY
05	F3	e^x	e^{-x}		Вызов из Р3 значения e^x
10	XY	e^{-x}	e^x		
11	—	e^x-e^{-x}	e^x	—	Вычисление e^x-e^{-x}
12	2	2	e^x-e^{-x}		Вызов в РХ цифры 2
13	÷	$\frac{e^x-e^{-x}}{2}$	e^x-e^{-x}		Вычисление значения sh$x$
14	С/П	$\frac{e^x-e^{-x}}{2}$	e^x-e^{-x}		Останов, индикация

Более короткое решение достигается, если перевести зна-чение e^{-x} в РY (команда ↑), вызвать в РХ значение e^x, поменять их местами (команда XY) и выполнить вычи-тание.

Заключительные шаги программы обеспечивают деле-ние e^x-e^{-x} на два и останов вычислений для индикаци результата (команда С/П). Так как запись в РХ циф-ры 2 следует после команды «Вычитание», перевод чис-ла e^x-e^{-x} в РY происходит автоматически.

Дополнительная возможность улучшения программы состоит в устранении команды Р2, оказавшейся лишней. Кроме того, если значение e^x сразу после его вычисле-ния поместить в РY (команда ↑), то после определе-ния значения e^{-x} можно выполнять вычитание без до-полнительных операций. Таким образом, оказываются излишними команды Р3, F3 и XY, а команда ↑ займет в программе новое место. В улучшенном варианте про-граммы (табл. 3.3) не используются регистры памяти и сокращено число программных шагов.

Программа для вычисления sh$x$ в Б3-34 отличается лишь обозначением соответствующих команд (табл

Таблица 3.3

Адрес	Команда	РХ	РY	Комментарий
00	$\text{Р}e^x$	e^x		Вычисление e^x
01	↑	e^x	e^x	Запись e^x в РY
02	$\text{F}1/x$	e^{-x}	e^x	Вычисление e^{-x}
03	—	e^x-e^{-x}	e^x	Вычисление разности e^x-e^{-x}
04	2	2	e^x-e^{-x}	Ввод цифры 2
05	÷	$\frac{e^x-e^{-x}}{2}$	e^x-e^{-x}	Вычисление sh x
10	С/П	$\frac{e^x-e^{-x}}{2}$	e^x-e^{-x}	Останов, индикация

3.4). Однако при ее составлении могут возникнуть и другие варианты, обусловленные наличием бóльшего числа регистров в операционном стеке. Так, например, вычисление значения e^{-x} (команда $\text{F}1/x$) уже на втором шаге программы обеспечит запоминание e^x в РХ1. Команда ↑ будет исключена, однако нахождение разности e^x-e^{-x} потребует ввода команд FBx и XY. Регистры Z, T, РП в данном случае не используются.

Оформление программы. После того как программа составлена и проверена, ее целесообразно представить в виде, удобном для дальнейшего использования. Табл. 3.1 для этой цели неудобна. Она громоздка и содержит много вспомогательной информации, которая при повторном применении программы не понадобится: это дан-

Таблица 3.4

Адрес	Команда	X	Y	X1	Комментарий
00	$\text{F}e^x$	e^x		x	Вычисление e^x
01	↑	e^x	e^x	x	Ввод e^x в РY
02	$\text{F}1/x$	e^{-x}	e^x	e^x	Вычисление e^{-x}
03	—	e^x-e^{-x}		e^{-x}	Вычисление e^x-e^{-x}
04	2	2	e^x-e^{-x}	e^{-x}	Ввод цифры 2
05	÷	sh x		2	Вычисление sh$x$
06	С/П	sh x		2	Останов, индикация

Таблица 3.5

Десятки	Единицы					
	0	1	2	3	4	5
0	Рe^x 33	↑ 06	F1/x 45	— 86	2 24	÷ 36
1 ... 9	С/П 78					

ные о состоянии регистров стека и РП, комментарии. Для того чтобы воспользоваться готовой программой, необходимо знать: собственно программу, т. е. последовательность вводимых в ОЗУ МК команд; коды команд программы, нужные для контроля правильности записи команд в ячейки ОЗУ; адреса записи отдельных команд, знание которых существенно упрощает поиск нужного участка программы, позволяет легко изменить адреса переходов, находить ошибки, связанные с пропуском команды, и т. д. Перечисленные данные удобно представить в виде таблицы. В качестве примера в табл. 3.5 [39] и 3.6 записаны программы вычисления sh$x$. Отличие табл. 3.5 (программа для Б3-21) и 3.6 (программа для Б3-34) отражает различную организацию памяти этих МК. Ячейки таблиц соответствуют ячейкам памяти ОЗУ программ. При записи более сложной программы число граф в таблице увеличивается и ограничивается емкостью памяти. Номер ячейки памяти определяется номерами пересекающихся строки и столбца, причем номер строки дает десятки, а номер столбца — единицы адреса ячейки памяти. В соответствующие ячейки таблицы записываются команды программы и их коды.

Таблица 3.6

Десятки	Единицы									
	0	1	2	3	4	5	6	7	8	9
0	Fe^x 16	↑ 0E	F1/x 23	— 11	2 02	÷ 13	С/П 50	—	—	—
1 ... 9										

Возможна и более компактная форма записи [27—33], в которой для Б3-21 в строку записывается 12 команд (две «страницы»), а для Б3-34 — десять. Коды команд в этой записи отсутствуют. Пример подобного оформления представляют собой программы 3.1/21 и 3.1/34 для вычисления sh$x$ (в обозначении программ принята тройная нумерация: первая цифра — номер главы, вторая — порядковый номер программы, третья — тип МК):

Программа 3.1/21

Рe^x ↑ F1/x — 2 ÷ С/П

Программа 3.1/34

Fe^x ↑ F1/x — 2 ÷ С/П

Такая форма записи программ используется в дальнейшем изложении.

3.2. Запись и отладка программ

Подготовка к реализации программы. Для записи составленных программ в ОЗУ МК необходимо выполнить действия, порядок которых представлен в табл. 3.7 применительно к программе вычисления sh$x$ (3.1/21 и 3.1/34).

Для того чтобы воспользоваться записанной в ОЗУ программой, необходимо знать, какие операции должны быть выполнены пользователем, различаются ли его действия при первом и последующих циклах вычислений и где размещаются результаты. Необходимо знать и область задания исходных данных.

Эти сведения о применении программы удобно представлять в виде инструкции. Так, при вычислении sh$x$ по программам 3.1/21 и 3.1/34 достаточно знать, что исходное значение x должно быть перед пуском программы записано в РХ и что перед началом вычислений производится очистка программного счетчика (В/О). Инструкция может быть записана так: x; В/О; С/П.

Если исходных данных несколько, инструкция записывается таким образом: А Р2 (П2); В Р3 (П3); x_1 Р4 (П4); x_2; В/О; С/П; y_0=<РХ>; y_1=<Р5>; y_2=

Таблица 3.7

Режим МК перед подачей команды	Команда МК Б3-21	Команда МК Б3-34	Комментарий
Работа	В/О	В/О	Установка программного счетчика на нулевой адрес
	РРП	FПРГ	Перевод МК в режим «Программирование»
Программирование	Pe^x ↑ F1/x — 2 ÷ С/П	Fe^x ↑ F1/x — 2 ÷ С/П	Ввод программы в ОЗУ МК. На индикаторе отражаются коды вводимых команд
	РРР	FАВТ	Перевод МК в режим «Работа»

=<Р6>. Это означает, что исходные константы А и В заносятся в Р2 и Р3; переменная x_1 заносится в Р4, переменная x_2 заносится в РХ, осуществляется очистка программного счетчика и пуск. Результаты y_0, y_1 и y_2 размещаются в регистрах РХ, Р5 и Р6. Если расположение результата не оговорено в инструкции, подразумевается, что он помещается в РХ.

При необходимости инструкция может быть дополнена словами, например, А Р2 (П2); x_0; В/О; С/П. При вычислении $y_1(x_1, \text{А})$ — x_1; В/О; С/П. Это означает, что в процессе вычислений исходное значение А в Р2 сохраняется. Если возврат к первой команде программы при повторном вычислении автоматизирован, то для вычисления y_1 будет записано: x_1; С/П.

Сложная программа может состоять из нескольких самостоятельных частей, переход между которыми должен выполнить пользователь с помощью команды БП с соответствующим адресом. Например, в МК сначала вводится массив данных x_i, затем массив данных y_i, после чего данные обрабатываются. Предполагается, что объем массивов может быть разным и переход ко второй и третьей частям программы автоматизировать нельзя. Тогда инструкцию можно записать так: В/О; x_i С/П; БП (адрес перехода); y_i С/П; БП (адрес перехода); С/П.

Отсутствие сведений о повторных вычислениях говорит об одинаковости действий в первом и последующих случаях применения программы.

Отладка программы. Прежде чем работать с составленной программой, ее необходимо проверить. С этой целью вычисляют результат при некоторых контрольных значениях исходных данных, которые выбираются так, чтобы результат вычислений был очевиден. Например, при $x=0$ имеем $\text{sh}x=0$. Однако в некоторых случаях подстановка, которая значительно упрощает используемое математическое выражение, может не выявить ошибку. Поэтому для проверки программы целесообразно сравнить результат, полученный с помощью МК для некоторого произвольного значения x с его известным значением. В справочниках по применению МК обычно приводятся контрольные решения.

Контрольное решение является первым шагом проверки программы. Правильность полученного результата позволяет переходить к непосредственному использованию программы.

Возможный порядок действий пользователя для обнаружения ошибки представлен в табл. 3.8. Для выявления ошибки удобно повторить контрольное решение пошагово (табл. 3.8, п. 2). Для этого следует установить программный счетчик на нулевой адрес и ввести в МК контрольные исходные данные, последовательно подавать команду пошагового прохождения программы ПП, после выполнения каждой команды сравнивать содержимое РХ в табл. 3.3, 3.4 и на индикаторе МК. При этом может быть обнаружен шаг n, начиная с которого содержимое РХ перестает соответствовать таблице. Причина может состоять в неверном вводе команды в ОЗУ программ. Для проверки необходимо, не подавая команду В/О, перевести МК в режим программирования и проверить код последней выполненной команды. Если этот код не соответствует требуемому, подать команду $\overleftarrow{\text{ШГ}}$ и ввести в ОЗУ правильную команду. При этом следует помнить, что изменение адреса М в составных командах требует повторного ввода и смысловой части команды. Далее необходимо повторить контрольное решение.

Если команда на шаге n введена в ОЗУ верно, необходимо проверить правильность ее использования, а также правильность описания работы МК при ее выполнении.

Таблица 3.8

Операция	Цель операции	Результат операции	Действия пользователя
1. Вычисления с использованием контрольных исходных данных в автоматическом режиме	Проверка правильности программы	Программа верна	Оформление программы в компактном виде
		Программа неверна	Переход ко второму этапу
2. Пошаговое прохождение программы	Определение шага n, на котором проявляется ошибка	Ошибка проявилась на шаге n	Проверка правильности ввода данной команды в ОЗУ; проверка правильности использования команды на шаге n; переход к третьему этапу
3. Пошаговое прохождение программы до шага n—1 и проверка содержимого регистров	Выявление регистра, в котором после шага n—1 информация записана неверно	Содержимое регистров соответствует таблице	Проверка правильности описания работы МК на шаге n в таблице с учетом возможных особенностей и устранение ошибки; например: исправление ошибки в программе и замена команды на шаге n; включение в программу на шаге n команды «Нет операции»

Продолжение табл. 3.8

Операция	Цель операции	Результат операции	Действия пользователя
		Содержимое одного из регистров не соответствует таблице	Переход к четвертому этапу
4. Пошаговое прохождение программы с проверкой на каждом шаге содержимого регистра, в котором обнаружена ошибка	Определение команды программы, после которой возникает ошибка	Ошибка обнаружена на шаге k	Устранение ошибки: проверка правильности ввода данной команды; проверка правильности описания работы МК при выполнении данной команды; проверка правильности использования данной команды

Следующий этап поиска ошибки — прохождение программы до шага $n-1$, что можно выполнить автоматически, временно включив в программу на шаге n команду С/П. Проверка содержимого регистров, участвующих в выполнении следующей команды (шаг n), позволяет осуществить поиск той команды, после которой в одном из регистров оказались неверные данные.

Несовершенство первых выпусков МК может обусловить возникновение ошибок, выявление которых весьма затруднено. Устранить эти ошибки в общем случае можно, сдвигая в ОЗУ всю программу или ее части (ввод команды «Нет операции»), либо дублируя некоторые команды пересылки в месте возникновения ошибки. Повторение контрольного решения после исправления ошибки обязательно.

Если программа применяется не впервые, т. е. имеется уверенность в ее правильности, ошибка может появиться лишь при вводе команд в ОЗУ.

Когда программа достаточно сложная, ее отладка может осуществляться по частям. С этой целью выделяют функционально законченный блок, после которого вместо следующей команды программы вводят команду останова. Отладив первый блок, изъятую команду восстанавливают и переходят к отладке следующего блока и т. д.

3.3. Составление разветвляющихся и циклических программ

Программы с разветвлениями и циклом. Разветвляющиеся и циклические программы организуются с использованием команд переходов. Особенности составления программ, содержащих такие команды, рассмотрим на примере вычисления $n!$ при целых n.

Схема алгоритма решения приведена на рис. 3.2. Поясним ход рассуждений при ее составлении. Поскольку величина $n!$ определена для $n \geqslant 0$, в программе необходимо предусмотреть проверку этого условия. При $n<0$ вычисления должны быть остановлены (блоки *1* и *2*), причем целесообразно зафиксировать факт некорректной операции.

В случае, когда $n=0$, имеем $n!=1$. Проверка условия $n=0$ осуществляется вторым логическим блоком (блок *3*). Когда $n>0$, значение $n!$ вычисляется по фор-

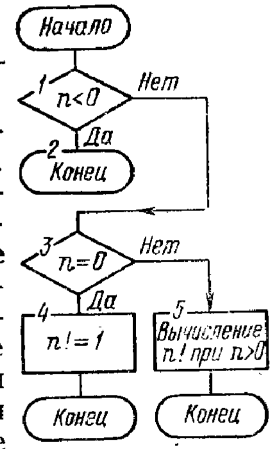


Рис. 3.2. Алгоритм вычисления $n!$

муле $n!=n(n-1)...1$. Обратимся к алгоритму организации цикла (рис. 1.6*а*). Подготовка цикла заключается в вычислении значения очередного сомножителя произведения $(n-k)$, после чего необходимо проверить его на равенство нулю. Если $(n-k)>0$, вычисляется тело цикла, представляющее собой определение нового значения произведения $n(n-1)...(n-k)$. В том случае, если $(n-k)=0$, вычисленное на предыдущем цикле произведение представляет собой искомое значение $n!$.

Разветвления в составляемом алгоритме выполняются с помощью команд условных переходов. Программа, реализующая данный алгоритм для Б3-34, приведена в табл. 3.9. Первая команда программы (адрес 00) осуществляет первую проверку по схеме алгоритма. При выполнении условия $n<0$ управление передается команде по адресу 02. Эта команда должна обеспечить останов вычислений и высветить на индикаторе сигнал ошибки. Для этого введем команду F $\sqrt{\ }$, т. е. искусственно создадим некорректную операцию при $n<0$.

Адрес перехода при $n \geqslant 0$ должен быть записан в программе сразу после команды проверки условия (по адресу 01). Следующую свободную ячейку памяти нужно использовать для записи команд, продолжающих вычисления при невыполнении проверяемого условия. Таким образом, по адресу 01 можно записать число 03.

Аналогично составляется участок программы с адреса 03 по адрес 06.

Продолжение вычислений связано с реализацией функции блока *5* (рис. 3.2). Воспользуемся формулой $n!=n(n-1)...(n-k)...1$. Выделим некоторый промежуточный результат $n(n-1)...(n-k)$. Чтобы его получить, необходимо знать предыдущее произведение $n(n-1)...$ $...(n-k+1)$ и новый сомножитель $(n-k)$. Так как новый сомножитель можно получить из последнего известного сомножителя $n-k+1$, то для вычисления нового произведения достаточно помнить предыдущее произве-

Таблица 3.9

Адрес	Команда	КОП	РХ	РY	РП	Комментарий
00	FX<0	5[	n			Проверка условия $n<0$
01						Место для адреса перехода при $n<0$
02	F√	21	ERROR			Искусственно созданная некорректная операция для останова и индикации ошибки
03	FX=0	5E	n			Проверка условия $n=0$
04						Место для адреса перехода
05	1	01	1			$0!=1$
06	С/П	50	1			Останов, индикация
07	П0	40	n		n=<P0>	Начальное распре-
08	П1	41	n		n=<P1>	деление памяти
09	ИП1	61	$n-k$			Вызов в РХ текущего значения $n-k$
10	1	01	1	$n-k$		Формирование нового сомножителя и проверка его на равенство нулю.
11	—	11	$n-k-1$			
12	FX=0	5E	$n-k-1$			
13						Место для адреса перехода
14	ИП0	60	$n!$			При $n-k-1=0$ произведение $n(n-1)\dots(n-k)=n!$
15	С/П	50	$n!$			Останов, индикация
16	П1	41	$n-k-1$			Запись в P1 нового текущего значения $n-k-1$
17	ИП0	60	$n(n-1)\dots(n-k)$	$n-k-1$		Вызов в РХ текущего значения произведения, вычисление нового произведения и запись его в P0
18	×	12	$n\dots(n-k-1)$			
19	П0	40	$n\dots(n-k-1)$			
20	БП	51	$n\dots(n-k-1)$			Безусловный переход к началу цикла
21						Место для адреса перехода

дение и его последний сомножитель. Отведем для них регистры P0 и P1 соответственно. На первом шаге вычислений в эти регистры согласно используемой формуле записано число n (команды с адресами 07 и 08).

Прежде чем вычислить новое произведение, надо сформировать новый сомножитель (команды с адресами 09—11), запомнить его значение (адрес 16), вычислить и запомнить новое произведение (адреса 17—19) и перейти на команду, начинающую описанный цикл, по адресу 09.

Таким образом, вычисления следует продолжать до тех пор, пока последний сомножитель произведения станет равен 1. Для того чтобы остановить при этом вычисления, необходимо проверить, не равен ли нулю сомножитель $n-k$ (команды с адресами 12—15).

Адреса переходов вписываются в соответствующие места программы после завершения программирования (программа 3.2/34).

Программа 3.2/34

FX<0	03	F√	FX=0	07	1	С/П	П0	П1	ИП1
1	—	FX=0	16	ИП0	С/П	П1	ИП0	×	П0
БП	09								

Для Б3-21 программа составляется аналогичным образом (программа 3.2/21)

Программа 3.2/21

PX<0	0	F√	PX=0	F1	1	С/П	P2	1	—	PX=0	P×
F2	С/П	↑	F2	×	P2	XY	БП	PXY			

Инструкция для обеих программ: n В/О С/П.

Использование команд организации циклов. В Б3-34 имеются специальные команды FL0 M; FL1 M; FL2 M; FL3 M, которые называются командами организации цикла. Эти команды можно заменить несколькими другими командами МК:

$$\text{FL0M} \equiv \text{ИП0; 1; —; П0; FX} = 0\ \text{M},$$

где M — адрес перехода. Проиллюстрируем эффективность применения команды организации цикла на примере программы вычисления $n!$.

В программе 3.2/34 фрагмент для вычисления $n!$, начиная с адреса 09 по адрес 13, дополненный командой по адресу 16, точно соответствует фрагменту, описывающему работу МК при выполнении команды FL1 M. Таким образом, этот фрагмент, содержащий пять команд, может быть заменен одной командой. В связи со сдвигом последующей части программы должен быть изменен адрес перехода, записанный в программе 3.2/34 в ячейку 13 (прогр. 3.3/34).

Программа 3.3/34

FX<0	03	F√	FX=0	07	1	С/П	П0	П1	FL1
13	ИП0	С/П	ИП1	ИП0	×	П0	БП	09	

Если теперь учесть, что выполнение команды FL1 13 не сопровождается изменением содержимого регистров операционного стека, то использование регистра Р0 оказывается излишним (прогр. 3.4/34).

Программа 3.4/34

FX<0	03	F√	FX=0	07	1	С/П	П1	FL1	11
С/П	ИП1	×	БП	08					

Если отказаться от проверки условий $n<0$ и $n=0$ (ввод в МК числа $n\leqslant 0$ возможен лишь при использовании программы для вычисления факториала в сочетании с другой программой), то окончательный вариант займет всего восемь шагов (программа 3.5/34).

Программа 3.5/34

П1	FL1	04	С/П	ИП1	×	БП	01

Инструкции для всех программ вычисления $n!$ одинаковы: n В/О С/П.

Организация подпрограмм. Если в программе некоторая последовательность команд повторяется несколько раз, например, при неоднократном вычислении некоторой функции с разными исходными данными, то обычно целесообразно оформление этих вычислений в виде подпрограммы. Подпрограмма записывается в свободных ячейках ОЗУ после основной программы. Для перехода к подпрограмме используется составная команда ПП М, где М — адрес перехода, т. е. адрес первой команды подпрограммы. Адрес указывается так же, как для условных и безусловных переходов. При переходе к подпрограмме адрес выхода из основной программы автоматически запоминается в специальном стеке возврата (см. § 2.1). Подпрограмма должна заканчиваться командой В/О, которая и обеспечивает обращение к указанному стеку и возврат к основной программе по хранящемуся в нем адресу. Стек возврата содержит пять регистров, что обес-

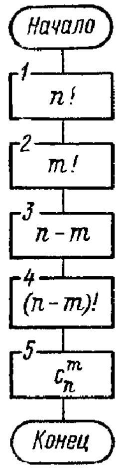


Рис. 3.3. Алгоритм вычисления C_n^m

печивает возможность обращения к подпрограмме внутри подпрограммы с глубиной обращений, равной пяти.

Выделять часть команд как подпрограмму целесообразно, когда это обеспечивает экономию памяти. В общем случае эффективность использования подпрограммы можно оценить количественно. Пусть n — число команд в повторяющемся участке программы, N — число обращений к подпрограмме, т. е. число повторений в программе одинаковой комбинации команд. Тогда число шагов подпрограммы будет $n+1$ (дополнительная команда возврата В/О). Для обращения к подпрограмме используются два шага программы: ПП М. Введение подпрограммы будет целесообразно, если $2N+n+1<Nn$. Если некоторый участок программы повторяется дважды ($N=2$) то использование подпрограммы целесообразно уже при $n=3$. Чем больше N и n — тем больше экономится ячеек памяти.

Рассмотрим использование подпрограммы на примере вычисления числа сочетаний $C_n^m = n!/m!(n-m)!$. Для упрощения откажемся от проверки условий $n\geqslant 0$, $m\geqslant 0$, $(n-m)\geqslant 0$, полагая, что они выполняются. Схема алгоритма решения изображена на рис. 3.3. Троекратное обращение к вычислению факториала позволяет оформить это вычисление как подпрограмму. Программа вычисления функции C_n^m для Б3-34 записана в табл. 3.10. Начиная с ячейки памяти с адресом 18 записывается подпрограмма вычисления факториала (см. табл. 3.9). Вычисление C_n^m в Б3-21 и Б3-34 обеспечивается программами 3.6/21 и 3.6/34.

Программа 3.6/21

F2	↑	F3	—	ПП	3	Р4	F3	ПП	3	Р3	F2
ПП	3	↑	F3	÷	↑	F4	÷	С/П	РХ=0	F4	1
В/О	Р5	1	—	РХ=0	Р/—/	F5	В/О	↑	F5	×	Р5
XY	БП	Р,									

Таблица 3.10

Адрес	Команда	КОП	РХ	РY	РZ	РП	Комментарий
						$n=\langle \text{Р0}\rangle$ $m=\langle \text{Р1}\rangle$	Исходное состояние
00	ИП0	60	n				Вычисление $n-m$ и переход к подпрограмме для вычисления $(n-m)!$
01	ИП1	61	m	n			
02	—	11	$n-m$				
03	ПП	53	$n-m$				
04	18	18	$n-m$				Адрес первой команды подпрограммы
05	П2	42	$(n-m)!$			$(n-m)!=\langle \text{Р2}\rangle$	Вычисление $n!$
06	ИП1	61	n				
07	ПП	53	n				
08	18	18	n				
09	П1	41	$n!$			$n!=\langle \text{Р1}\rangle$	
10	ИП0	60	m				Вычисление $m!$
11	ПП	53	m				
12	18	18	m				
13	ИП1	61	$n!$	$m!$			
14	ИП2	62	$(n-m)!$	$n!$	$m!$		Вычисление C_n^m
15	×	12	$n!\,(n-m)!$	$m!$			
16	÷	13	C_n^m				
17	С/П	50	C_n^m				Останов, индикация

Программа 3.6/34

ИП0	ИП1	—	ПП	18	П2	ИП1	ПП	18	П1
ИП0	ПП	18	ИП1	ИП2	×	÷	С/П	FX=0	22
1	В/О	П3	П4	ИП4	1	—	FX=0	31	ИП3
В/О	П4	ИП3	×	П3	БП	24			

Инструкция к программе 3.6/21: $n\geqslant 0$; $m\geqslant 0$; $n\geqslant m$; n Р2 m Р3В/О С/П. Инструкция к программе 3.6/34: $n\geqslant 0$, $m\geqslant 0$; $n\geqslant m$; n П0; m П1 В/О С/П. Как видно из инструкций, в этих программах предполагается, что $n\geqslant 0$, $m\geqslant 0$, $n\geqslant m$, и выполнение этого условия в отличие от табл. 3.9 не проверяется.

Использование подпрограммы для вычисления C_n^m уменьшает программу в Б3-21 на 36 шагов, в Б3-34— на 38.

Вспомогательные операции. Для удобства работы с программой целесообразно максимально возможно сокращать число операций, выполняемых пользователем вручную, в том числе сокращать число команд программы, т. е. число нажимаемых клавиш при ее вводе в ОЗУ. Для пользователя оказывается наиболее важным сокращение числа ручных операций, выполняемых в режиме «Работа». Эти операции описываются инструкцией к программе и включают в себя установку начального адреса перед началом вычислений, ввод используемых констант, ввод исходных значений аргументов и т. д. При наличии свободных ячеек памяти ОЗУ названные операции могут быть введены в программу и выполняться автоматически. Увеличение времени на ввод и выполнение удлиненной таким образом программы не существенно, ибо оно всегда меньше времени, затрачиваемого пользователем на ручное выполнение соответствующих операций. Кроме того, это всегда ведет к упрощению инструкции к программе.

Простейший пример вспомогательной операции — организация автоматического возврата к началу программы. Рассмотренный в § 3.2 порядок пользования программой предусматривает подачу команды В/О перед каждым новым вычислением. По этой команде обнуляется программный счетчик, т. е. устанавливается на первую команду программы. В некоторых случаях команда В/О может не подаваться, тогда программный счетчик будет просматривать оставшиеся ячейки ОЗУ и циклически выйдет на первую команду. Однако это удлинит работу, а если в неиспользуемых ячейках ОЗУ окажется какая-то команда, то программа будет выполнена неверно. Поэтому возврат к началу программы всегда целесообразен, и он может выполняться автоматически. Для этого после окончания программы (команда С/П) вводят команду безусловного перехода к нулевому адресу. Этот фрагмент программы в Б3-21 будет выглядеть так: ...С/П БП Р0; а в Б3-34: ...С/П БП 00. Включив в программу перечисленные команды, достаточно подать один раз команду В/О после записи программы в ОЗУ, а затем обновлять только исходные данные. Автоматический возврат возможен и при записи программы не с нулевой ячейкой ОЗУ, для этого необходимо в фрагменте возврата указать соответствующий адрес.

Значительные возможности улучшения программы заключены в автоматизации введения исходных данных, включающих константы, содержащиеся в математическом выражении для вычислений, и исходные значения аргументов. Например, распределение исходных данных в памяти МК удобнее и надежнее оформить в виде программы, чтобы пользователю при ее применении нужно было подавать только команду С/П.

Значительно уменьшить число ручных операций может программное формирование значений констант, а также очистка регистров.

В некоторых случаях повторное применение программы связано с изменением значения аргумента на фиксированное значение, например, для построения графика значений функции y для значений аргумента x_i с шагом Δx. В этих условиях возможно программное формирование нового исходного данного для следующего вычисления. Если значение x_i помещено в Р2, а значение Δx в Р3, то команды обеспечивающие формирование x_{i+1} и его запоминание в Б3-21 будут следующие: F2; ↑; F3; +; Р2; в Б3-34; ИП2; ИП3; +; П2. После формирования x_{i+1} обеспечивается безусловный переход к началу программы.

При организации вспомогательных операций так же, как и при составлении основной программы, следует руководствоваться принципом минимальной длины программы.

Применение команд косвенной адресации. Как указывалось в § 2.2, команда с косвенной адресацией, выполняя в программе те же функции, что и команда с прямой адресацией, занимает одну ячейку памяти вместо двух. Однако применение команд косвенной адресации, сокращая необходимое число ячеек памяти ОЗУ, ведет к дополнительному расходу РП и увеличению числа операций, связанных с вводом в РП соответствующих адресов.

Команды косвенной адресации вносят новые возможности в работу МК. В связи с тем, что адрес перехода записывается не в самой программе, а в некотором РП, возможно в процессе решения задачи изменять этот адрес и тем самым изменять последовательность прохождения программы.

Модификация числа, находящегося в регистре, в определенных случаях может быть использована для выполнения операций вычитания или прибавления единицы к целым положительным числам. Например, две команды КИП4; ИП4 заменяют в программе четыре команды ИП4; 1; +; П4, что весьма удобно при организации счетчика циклов.

Такое применение команд косвенной адресации требует дополнительного внимания, ибо косвенное обращение к регистру (КИПN) обеспечит перемещение информации в операционном стеке и вызов в РХ числа из некоторого регистра.

Использование команд косвенного перехода по модифицированному адресу к подпрограмме, безусловного или условного косвенного перехода обеспечивает возможность создания циклов в тех случаях, когда очередное повторение определенной последовательности команд связано с выполнением новой подпрограммы или переходом к новому адресу.

Адрес перехода в регистре, указанном в команде косвенной адресации, может быть задан некорректно: отрицательное число, дробное число или число, соответствующее номеру отсутствующего в МК РП. Однако команды вида КИПN в этой ситуации могут быть использованы, поскольку их выполнение не ведет к каким-либо переходам в программе, в то же время изменения, происходящие с числом в регистре N, могут использоваться для сокращения длины программы, либо для получения оригинальных решений.

Строго говоря, модификация числа, находящегося в РП, номер которого входит в команду косвенной адресации, выполняется для всех регистров. Записанное в регистре положительное число при исполнении команды косвенной адресации всегда воспринимается как адрес, а адрес не может быть дробным. Независимо от того, должно измениться содержимое регистра на единицу или нет, предварительно оно округляется с точностью до ненулевой целой части. Это свойство можно использовать для выделения целой части числа (табл. 3.11).

В том случае, когда в регистр записано положительное число с нулевой целой частью, в регистрах 0—6 происходит нормализация этого числа с последующим вычитанием или прибавлением единицы в младшем разряде индикатора. При записи числа с нулевой целой ча-

Таблица 3.11

Исходное состояние регистра, указанного в команде	Поданная команда	Состояние регистра после подачи команд
5,50	КИП0	4
5,50	КИП4	6
5,50	КИП8	5
0,30	КИП0	$2,9999999 \cdot 10^{-1}$
0,9999999	КИП4	$9,9999991 \cdot 10^{-1}$
0,25	КИП7	0,25
—28,00	КИП9	—99999928
—2,85	КИП5	—99999993
—0,60	КИП6	$-6,0000001 \cdot 10^{-1}$
—0,056	КИП1	$-5,5999999 \cdot 10^{-2}$

стью в регистры Р7—РД подача команды косвенной адресации не изменит содержимого регистра.

Определенный интерес представляют случаи, когда в регистр, входящий в команду косвенной адресации, записано отрицательное число (см. табл. 3.11).

Таким образом, возможности использования команд косвенной адресации оказываются весьма широкими и порой неожиданными.

Глава четвертая

МЕТОДЫ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ РАБОТ НА МИКРОКАЛЬКУЛЯТОРАХ

4.1. Распределение памяти микрокалькуляторов

Возможности оптимизации распределения памяти. Существенным ограничением для применения МК являются небольшая емкость программного ОЗУ и малое число регистров памяти (РП), поэтому эффективность использования МК в значительной мере зависит от умелого их использования.

Возможны различные критерии оптимизации составляемых программ. Целью оптимизации распределения памяти может быть экономия числа операций или числа используемых РП. Многообразие вычислительных алгоритмов затрудняет разработку регулярных методов оптимизации распределения памяти программ. Рассмотрим некоторые из них.

Возможности сокращения числа операций при заданном входном языке МК заключаются в основном в использовании подпрограмм и

в применении методов косвенной адресации (см. § 3.3, 3.4). Большой интерес представляет изучение возможностей сокращения числа используемых РП, число которых в МК невелико.

Назначение РП состоит в хранении переменных, представляющих собой исходные данные и промежуточные результаты вычислений. По мере использования необходимость в последующем хранении переменных может устраняться, тогда соответствующий РП может переобозначаться, т. е. использоваться для хранения новой переменной. Поскольку при выполнении операций обработки в МК используются регистры операционных стеков, сокращение числа обращений к РП может быть достигнуто при максимальном использовании возможностей этих регистров.

Использование регистров операционных стеков. Среди операционных регистров РХ является регистром-аккумулятором: он предназначен для хранения операнда, и в него же помещается результат операции. Второй операционный регистр РY связан с РХ по принципу стека (см. гл. 2). Существующий порядок обмена между регистрами операционного стека достаточен для выполнения цепочечных вычислений, т. е. таких, когда результат предыдущей операции является операндом для последующей. В обобщенном виде алгоритм цепочечной операции описывается выражением

$$F_i\left\{\left[\left(\frac{(F_i(a_1) \pm F_i(a_2) \pm \ldots) F_i(b_1) F_i(b_2) \ldots}{F_i(c_1) F_i(c_2)}\right) \pm F_i(d_1) \pm \right.\right.$$
$$\left.\left. \pm F_i(d_2) \pm \ldots\right] F_i(e_1) \ldots\right\} \pm \ldots, \qquad (4.1)$$

где a; b; c; d; e — операнды, вводимые в РХ из РП или ПЗУ чисел в ручном или автоматическом режиме; F_i — символ любой одноместной операции.

Операционный стек из двух регистров обеспечивает выполнение в принципе сколь угодно длинной последовательности действий по формуле (4.1) с соответственно большим уровнем скобок.

Напомним, что в Б3-21 ввод нового операнда из РП требует предварительной подачи команды ↑ для перевода предыдущего результата в РY. При наборе числа с помощью клавиатуры после операции обработки этот перевод выполняется автоматически. В Б3-34 сдвиг чисел из РХ в РY происходит автоматически и при вызове операнда из РП. Команда ↑ всегда необходима для разделения двух первых операндов, если первая операция двухместная.

Легко убедиться, что вычисления по формуле (4.1) возможны без использования дополнительных РП только в случае, если они начинаются с определения суммы, стоящей в круглых скобках. Зависимость эффективности вычислений от их очередности относится к общим свойствам алгоритмов и является основой для поиска путей оптимизации программ.

В случаях когда вычисляемое выражение не может быть сведено к алгоритму с цепочечной структурой (4.1), требуются дополнительные регистры. В Б3-21 имеется только семь РП, поэтому они наиболее дефицитны. Использование регистров кольцевого стека, удобных для последовательного «откладывания» результатов предварительных вычислений, в некоторых случаях может быть затруднено (см. § 2.1).

Существенное увеличение возможностей и одновременно упрощение программирования вычислений достигается в Б3-34 усложнением структуры операционного стека. Основные правила передачи данных в регистрах приведены в табл. 2.2, 2.3. Использование лишь одного дополнительного регистра PZ обеспечивает возможность вычисления без обращения к РП выражения вида

$$F_i\left\{\left[\left(\frac{(F_i(A_1) \pm F_i(A_2) \pm \ldots)F_i(B_1)\,F_i(B_2)\ldots}{F_i(C_1)\,F_i(C_2)\ldots}\right) \pm F_i(D_1) \pm \right.\right.$$

$$\left.\left. \pm F_i(D_2) \pm \ldots\right] F_i(E_1)\ldots\right\} \pm \ldots, \tag{4.2}$$

где F_i — символ одноместной операции; A, B,... — блоки, допускающие возможность вычисления по цепочечной схеме [см. формулу (4.1)].

После проведения вычислений $F_i(A_1)$ результат помещается в PX. В остальных регистрах стека — нули. Нулевой результат в PY объясняется тем, что при последней двухместной операции по вычислению $F_i(A_1)$ результат помещается в PX, а в PY переписывается ноль из PZ. Если первой операцией вычисления $F_i(A_2)$ является двухместная операция, тогда введение операндов a_2 и b_2 переместит результат $F_i(A_1)$ в регистр PZ. Любые операции в цепочечной схеме (4.1) по вычислению $F_i(A_2)$ используют лишь PX и PY. При этом после двухместной операции «отложенный» результат $F_i(A_1)$ перемещается в PY, а после введения двух операндов для следующей операции вновь сдвигается в PZ. После окончания вычисления $F_i(A_2)$ определяется сумма $F_i(A_1) \pm F_i(A_2)$ и т. д. Регистр T при этом не используется.

Для иллюстрации использования PT рассмотрим пример вычисления выражения

$$y = (a_1 b_1 + a_2 b_2)(a_3 b_3 + a_4 b_4). \tag{4.3}$$

Для вычисления каждого выражения в скобках требуется использование трех регистров, а один регистр необходим для запоминания результата вычисления одной из скобок во время вычисления второй. Следовательно, выражение (4.3) более сложное, чем (4.2). Операционный регистр в МК Б3-34 обеспечивает выполнение требуемых вычислений без использования дополнительных РП. Дальнейшее усложнение связано, например, с вычислением выражения

$$y = [(a_1 b_1 + a_2 b_2)(a_3 b_3 + a_4 b_4)] + [(m_1 n_1 + m_2 n_2)(m_3 n_3 + m_4 n_4)]. \tag{4.4}$$

Вычисление y в формуле (4.4) требует использования дополнительного РП для хранения результата вычисления в квадратной скобке, поскольку для вычисления второй квадратной скобки требуются все четыре регистра стека. Исключить необходимость использования РП в данном случае можно путем преобразования формы записи выражения (4.4), например, раскрыв круглые скобки во втором слагаемом:

$$m_1 n_1 m_3 n_3 + m_2 n_2 m_3 n_3 + m_1 n_1 m_4 n_4 + m_2 n_2 m_4 n_4. \tag{4.5}$$

Выражение (4.5) относится к цепочечному типу (4.1), и для его вычисления требуются лишь два регистра стека. Однако, если для вычисления квадратной скобки в выражении (4.4) требовалось два сложения и пять умножений, то для вычисления формулы (4.5) нужно двенадцать умножений и три сложения, т. е. число программных шагов возросло более чем вдвое.

Таблица 4.1

Операция	Содержание регистров после операции				
	X1	X	Y	Z	T
Исходное состояние	0	A	0	0	0
F_1	A	F_1 (A)	0	0	0
FBx	A	A	F_1 (A)	0	0
F_2	A	F_2 (A)	F_1 (A)	0	0
FBx	A	A	F_2 (A)	F_1 (A)	0
F_3	A	F_3 (A)	F_2 (A)	F_1 (A)	0
FBx	A	A	F_3 (A)	F_2 (A)	F_1 (A)
F_4	A	F_4 (A)	F_3 (A)	F_2 (A)	F_1 (A)
$\Phi_1 = F_4 + F_3$	F_4 (A)	Φ_1	F_2 (A)	F_1 (A)	F_1 (A)

В соответствии с изложенным для экономии числа необходимых РП при программировании следует стремиться сделать преобразования, выделяющие максимально крупные блоки вычислений, «посильных» для используемого МК.

Дополнительные возможности операционного стека Б3-34 реализуются при использовании регистра предыдущего результата PX1, что обеспечивает вычисление до четырех одноместных операций F_i от аргумента A с последующим выполнением до трех двуместных операций над величинами $F_i(A)$: $\Phi[F_1(A);\ F_2(A);\ F_3(A);\ F_4(A)]$, где Φ — последовательность операций над содержимым регистров X, Y, Z, T операционного стека.

Перемещение данных в стеке при использовании PX1 представлено в табл. 4.1.

Кольцевой сдвиг данных в стеке служит средством «откладывания» результата вычислений A, полученного в PX, на некоторое число шагов программы. При выполнении команды F⟶ содержимое PX передается в PT и в PX1. Данные в остальных регистрах сдвигаются влево, как представлено в табл. 2.3. После такой операции нельзя производить ввод операнда в PX, поскольку содержимое PT будет утрачено («выдвинуто» из стека вправо). Поэтому наиболее удобно программировать вычисления так, чтобы после команды F⟶ следовала двухместная операция. Тогда операнд из PT перейдет в PZ, и станет возможным выполнение любых операций, требующих введения в PX не более одного операнда, вплоть до необходимости использовать «отложенный» результат.

Рациональное использование регистров памяти. Как отмечалось выше, алгоритмы вычислений не представляют однозначно программу вычислений. Отдельные операции алгоритма могут выполняться в различной последовательности, в результате чего изменяются требования к необходимой емкости памяти.

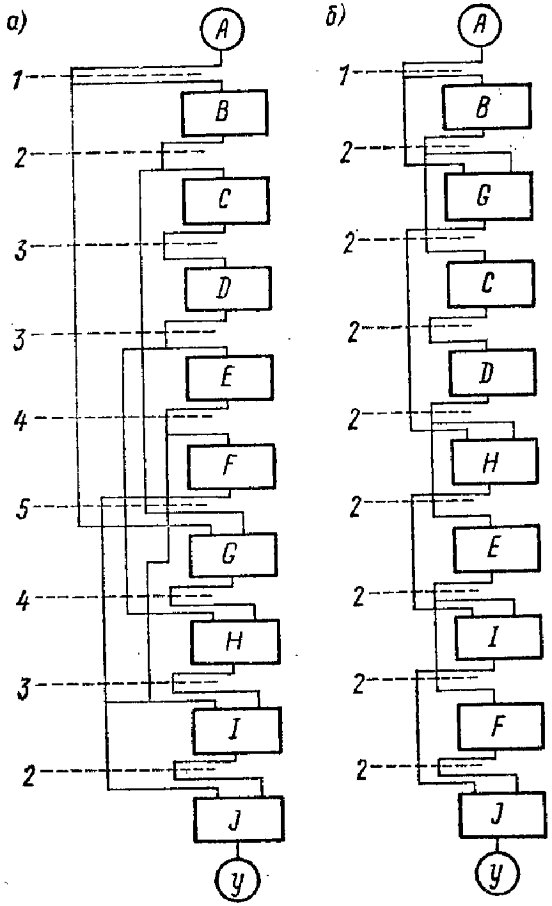
В универсальных ЭВМ распределение памяти машины производят специальные программы — компиляторы (трансляторы, интерпретаторы) [14]. В МК распределение памяти производится вручную, что требует более высокой квалификации пользователя, зато обеспечивает лучшие возможности оптимизации программ.

При программировании вычислений на МК для оптимизации распределения памяти (см. § 4.1) алгоритм вычислений следует разделить на блоки, допускающие вычисления без использования РП, и определить требования к передаче данных между этими блоками — информационные связи. При этом содержание выполняемых операций не имеет значения, важны лишь число и адреса источников входных и потребителей выходных данных. Схема алгоритма, отражающая только информационные связи операторов, называется операторной схемой. Существует достаточно общая теория построения и преобразования операторных схем с целью экономии ячеек памяти [10]. Проиллюстрируем методику оптимизации распределения памяти в МК на примере программирования вычислений по формуле

$$y=\left[\frac{\frac{\sin A}{A}(1-2\sin^2 A)+e^{-(1-\sin^2 A)}}{1+e^{(1-2\sin^2 A)}}\right], \qquad (4.6)$$

где $A=A(a_1;\ a_2;\ a_3...)$ — параметр, получаемый при предварительных вычислениях.

Операторная схема алгоритма (4.6) показана на рис. 4.1, где выделены блоки вычислений, соответствующие последовательному выполнению операций:



$$B(A)=\sin A;$$
$$C(B)=\sin^2 A;$$
$$D(C)=(1-2\sin^2 A);$$
$$E(D)=e^{-(1-2\sin^2 A)};$$
$$F(E)=1+e^{(1-2\sin^2 A)};$$
$$G(A;\ B)=A/B;$$
$$H(G;\ D)=GD;$$
$$I(H;\ E)=H+E;$$
$$J(I;\ F)=I/F.$$

Весь набор информационных связей называют связкой, а число линий в связке — шириной ее сечения [10]. Сечения связки изображены на рис. 4.1 штриховыми линиями, а ширина сечения обозначена цифрами. Из рисунка 4.1 видно, что реализация программы вычислений (4.6) по схеме рис. 4.1, *а* требует использования пяти РП.

Рис. 4.1. Операторная схема алгоритма

Возможность оптимизации программы основана на изменении порядка вычислений. Ясно, что наименьшая емкость памяти потребуется, если удастся как можно раньше использовать результаты вычислений в предыдущих блоках. Так, например, оператор *G* следует выполнять сразу после блока *B*. Преобразованная по такому принципу операторная схема алгоритма представлена на рис. 4.1, *б*. Максимальная ширина связи в этом случае равна двум.

Возможны случаи, когда укорочение одних связей потребует удлинения других и оптимизация программы не будет столь очевидной. Тем не менее и в сложной ситуации наглядность операторной схемы позволит избежать хотя бы грубых просчетов в распределении памяти или отыскать, например, единственный недостающий для реализации программы РП.

Использование составных программ. Используя рассмотренные методы, в некоторых случаях все же не удается разместить сложную программу в памяти МК. Тогда можно попытаться разделить ее на части. Введение этих частей в память осуществляется вручную, однако передача результатов вычислений от предыдущих частей к последующим осуществляется с помощью РП МК. Сложная программа, представленная в виде отдельных частей, называется пакетом программ [28].

Если нехватка памяти МК незначительна, можно эффективно использовать комбинирование вычислений в автоматическом и ручном режимах. При этом ручные вычисления производятся для определения значения функции от аргумента *x*, полученного предварительно при автоматических вычислениях. В наиболее жестких условиях все РП МК заняты, тогда возможности ручных вычислений определяются возможностями операционного стека МК.

Эти вычисления проводятся вручную после остановки автоматических вычислений, для чего в требуемых местах основной программы используются команды С/П.

4.2. Погрешности вычислений и методы их уменьшения

Анализ точности полученного результата является важной составной частью вычислительного процесса. Точность результата оценивается с использованием понятия погрешности.

Погрешность — это величина, характеризующая степень близости точных и приближенных значений получаемых величин. При анализе погрешностей используются различные их виды [13, 17] в зависимости от конкретных условий, например, абсолютная погрешность, относительная погрешность и т. д.

Абсолютная погрешность представляет собой разность между истинным значением вычисляемой величины x и ее приближенным значением $\hat{x}$: $\Delta x=\hat{x}-x$. Поскольку истинное значение x неизвестно, то под оценкой абсолютной погрешности понимают установление неравенства вида $|\hat{x}-x|\leqslant\Delta x_п$, где $\Delta x_п$ — так называемая предельная абсолютная погрешность.

Относительная погрешность определяется как отношение абсолютной погрешности Δx к модулю истинного значения x (иногда — к модулю приближенного значения $\hat{x}$): $\delta x=\Delta x/|x|\approx\Delta x/|\hat{x}|$. При оценке относительной погрешности пользуются понятием предельной относительной погрешности $\delta x_п$, которое удовлетворяет неравенству $|\delta x|\leqslant\delta x_п$.

Источниками погрешностей являются: конечная точность представления исходной информации (например, округление значений входных величин или конечная точность их измерений), особенности используемого приближенного численного метода, распространение погрешностей в процессе вычислений. В связи с этим различают инструментальные погрешности, методические погрешности, неустранимые или наследственные погрешности.

Инструментальные погрешности в основном связаны с округлением входных значений, промежуточных и итоговых результатов при выполнении отдельных операций в МК. Они зависят как от типа МК, так и от численного метода. При округлении отбрасываются младшие разряды чисел, значения которых либо не являются достоверными, либо выходят за разрядную сетку МК.

Существует три способа округления:

1. Несимметричное округление по дополнению. Младший сохраняемый разряд не изменяется, если старший из отбрасываемых меньше пяти, и увеличивается на единицу, если старший из отбрасываемых разрядов больше или равен пяти.

2. Симметричное округление по дополнению. Младший разряд не изменяется, если старший из отбрасываемых меньше пяти, и увеличивается на единицу, если старший из отбрасываемых разрядов больше пяти. Когда старший из отбрасываемых разрядов равен пяти, то четный младший сохраняемый разряд остается без изменения, а нечетный увеличивается на единицу.

3. Округление отбрасыванием. Младший сохраняющийся разряд не изменяется при любых значениях отбрасываемых разрядов.

Первые два способа округления обеспечивают более точный конечный результат в сравнении с простым отбрасыванием.

В различных модификациях МК используются различные способы округления. Для того чтобы определить алгоритм округления в используемом МК, можно выполнить операцию 2 : 3. В тех МК, где округление производится отбрасыванием, получим результат $10^{-1}\times\times6{,}6666666\cdot10^{-1}$. В тех МК, где округление учитывает значение девятой значащей цифры, результат будет равен $10^{-1}\cdot6{,}6666667$.

Очевидно, округление по дополнению возможно лишь тогда, когда внутренние регистры МК имеют большее число разрядов. Эти разряды называются скрытыми. Как правило, пользователь не имеет доступа к этим разрядам.

Ограниченное число разрядов МК и связанное с этим округление может привести к появлению таких результатов: $(1:3)\cdot3=10^{-1}\cdot9{,}9999999$ (округление отбрасыванием); $(2:3)\cdot3=10^{-1}\times\times2{,}0000001$ (округление по дополнению).

Программируемые МК обладающие большой емкостью памяти, дают возможность вычислять результат, число значащих цифр мантиссы которого существенно больше разрядности МК [32]. Такой способ повышения точности увеличивает время выполнения вычислений, причем выполнение каждого вида операции с повышенной точностью требует отдельного программного обеспечения. Такие программы занимают значительную емкость памяти МК [28].

Методические погрешности обусловлены тем, что многие задачи на МК решаются приближенно с использованием специальных численных методов (см. гл. 5). Это относится к тригонометрическим, логарифмическим, показательным и другим функциям. Ошибки вычислений этих функций могут быть существенно большие, чем ошибки округления. Например, произведение $4{,}01\times4{,}01\times4{,}01=64{,}481201$ с помощью МК с восьмиразрядными регистрами будет получено точно. В то же время использование функции x^y в Б3-34 дает результат 64,481180, а в Б3-21 — 64,481621.

В большинстве руководств по численным методам приводятся выражения, позволяющие оценить по крайней мере ориентировочно предельную методическую погрешность [13, 17, 41].

Наследственные погрешности — это погрешности результата вычисления, вызванные распространением или трансформацией через вычислительный алгоритм погрешностей исходных данных. Рассмотрим трансформацию погрешностей исходных данных в погрешность результатов каждого из четырех арифметических действий [13, 17].

1. Абсолютная погрешность алгебраической суммы приближенных чисел равна сумме абсолютных погрешностей слагаемых:

$$\Delta(x+y)=\Delta x+\Delta y.$$

2. Относительная погрешность суммы нескольких чисел одного знака заключена между наименьшей и наибольшей из относительных погрешностей слагаемых:

$$\delta y\leqslant\delta(x+y)\leqslant\delta x \text{ (считаем, что } \delta x>\delta y).$$

3. Относительная погрешность разности двух положительных чисел больше относительных погрешностей этих чисел, особенно, если эти числа близки между собой.

4. При умножении и делении приближенных чисел их относительные погрешности складываются:

$$\delta(xy)=\delta x+\delta y;\quad \delta(x/y)=\delta x+\delta y.$$

5. Абсолютная погрешность произведения или частного определяется по его относительной погрешности:

$$\Delta(xy)=|xy|\,\delta(xy);\quad \Delta(x/y)=|x/y|\,\delta(x/y).$$

В связи с этим можно рекомендовать следующие правила для уменьшения погрешностей вычислений [17]:

1. При сложении и вычитании длинной последовательности чисел следует сначала оперировать с наименьшими по модулю числами.

2. Следует избегать вычитания близких по значению чисел и сложения чисел, отличающихся на несколько порядков.

РАЗДЕЛ ВТОРОЙ

МЕТОДЫ, АЛГОРИТМЫ И ПРОГРАММЫ РЕШЕНИЯ ЗАДАЧ НА МИКРОКАЛЬКУЛЯТОРАХ

Глава пятая

АНАЛИЗ ФУНКЦИЙ И РЕШЕНИЕ УРАВНЕНИЙ

5.1. Интерполяция и аппроксимация функций

Пусть функция $y=f(x)$ задана в виде множества значений $\{y_i=f(x_i)\}$, определенных в дискретных равноудаленных точках x_i

$$x_i = x_0 + i\Delta x \ (i = 0,\dots, n),$$

где $n+1$ — число дискретных отсчетов функции; x_0 — начало отсчета аргумента x; Δx — шаг дискретизации.

Задача интерполяции состоит в определении значения функции $y=f(x)$ в некоторой промежуточной точке x, не совпадающей в общем случае ни с одним из узлов интерполяции x_i, $i=0,\dots, n$.

Как правило, задача интерполяции решается следующим образом. Выбирается набор так называемых базисных функций $\{\varphi_0(x),\dots, \varphi_N(x)\}$ и строится интерполирующая функция

$$P(x) = \sum_{j=0}^{N} a_j \varphi_j(x), \qquad (5.1)$$

коэффициенты a_j которой определяются из условия

$$P(x_i) = y_i \ (i = 0,\dots, n). \qquad (5.2)$$

После этого в качестве интерполированного значения функции $\hat{f}(x)$ в точке x принимается значение в этой точке интерполирующей функции $\hat{f}(x)=P(x)$, причем принимается $x_0 \leqslant x \leqslant x_n$.

Отметим, что интерполирующие функции могут использоваться как для экстраполяции назад $(x<x_0)$, так и для экстраполяции вперед $(x>x_n)$.

Существует целый ряд методов интерполяции [13, 41]. Рассмотрим два из них.

Интерполяционный многочлен Лагранжа N-й степени имеет вид

$$P_N(x) = \sum_{j=0}^{N} y_j \frac{(x-x_0)(x-x_1)\dots(x-x_{j-1})\times}{(x_j-x_0)(x_j-x_1)\dots(x_j-x_{j-1})\times} \to$$

$$\to \frac{(x-x_{j+1})\dots(x-x_N)}{(x_j-x_{j+1})\dots(x_j-x_N)}. \qquad (5.3)$$

Базисные функции в этом случае определяются выражением

$$\varphi_j(x) = \frac{(x-x_0)\dots(x-x_{j-1})(x-x_{j+1})\dots(x-x_N)}{(x_j-x_0)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_N)}. \qquad (5.4)$$

Поскольку каждая из этих функций $\varphi_j(x)$ равна нулю во всех узлах, кроме узла x_j $(\varphi_j(x_j)=1)$, то коэффициентами интерполяционного многочлена являются значения исходной функции: $a_j=y_j$, что соответствует выражению (5.3).

В качестве примера рассмотрим интерполяционные многочлены первой и второй степеней:

$$P_1(x) = \frac{y_0(x-x_1)-y_1(x-x_0)}{\Delta x}; \quad x\in[x_0, x_1]; \qquad (5.5)$$

$$P_2(x) = \frac{y_0(x-x_1)(x-x_2)-2y_1(x-x_0)(x-x_2)+}{2(\Delta x)^2} \to$$

$$\to \frac{+y_2(x-x_0)(x-x_1)}{2(\Delta x)^2}; \quad x\in[x_0, x_2]. \qquad (5.6)$$

Выражение (5.5) удобнее использовать в виде

$$P_1(x) = (1-p)y_0 + py_1, \qquad (5.7)$$

где $p=(x-x_0)/\Delta x$.

Программы 5.1 реализуют этот метод:

Программа 5.1/21

Р2	С/П	Р3	С/П	Р4	С/П	Р5	С/П	↑	F2	—	Р6
F3	↑	F2	—	F1/x	↑	F6	×	Р6	/—/	↑	1
+	↑	F4	×	Р7	F6	↑	F5	×	↑	F7	+
F7	С/П	БП	РXY								

Программа 5.1/34

П2	С/П	П3	С/П	П4	С/П	П5	С/П	ИП2	—
П6	ИП3	ИП2	—	F1/x	ИП6	×	П6	/—/	1
+	ИП4	×	П7	ИП6	ИП5	×	ИП7	+	П7
С/П	БП	08							

Инструкция к программам 5.1: В/О; набрать x_0 С/П; x_1 С/П; y_0 С/П; y_1 С/П; набрать x С/П; результат $\hat{f}(x)$ содержится в РХ и Р7. Если следующее значение x лежит в прежнем интервале, то набрать x и произвести пуск программы (С/П) без обновления других данных.

Другим распространенным методом интерполяции является метод Ньютона. В этом случае интерполяционный многочлен имеет вид

$$P_N(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \ldots + C_N(x - x_0)(x - x_1)\ldots(x - x_{N-1}). \qquad (5.8)$$

Коэффициенты C_j находятся из уравнений (5.2), которые преобразуются к следующей форме:

$$C_0 = y_0;$$
$$C_0 + C_1(x_1 - x_0) = y_1;$$
$$C_0 + C_1(x_2 - x_0) + C_2(x_2 - x_0)(x_2 - x_1) = y_2;$$
$$\cdots\cdots\cdots$$
$$C_0 + \ldots + C_N(x_N - x_0)(x_N - x_1)\ldots(x_N - x_{N-1}) = y_N.$$

Для определения коэффициентов C_j целесообразно использовать метод, основанный на применении правых конечных разностей [41].

Первая правая конечная разность определяется выражением

$$\Delta y_i = y_{i+1} - y_i.$$

Вторая конечная разность

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

Правая конечная разность порядка j определяется соотношением

$$\Delta^j y_i = \Delta^{j-1} y_{i+1} - \Delta^{j-1} y_i.$$

Можно показать, что коэффициенты интерполяционного полинома выражаются следующим образом:

$$C_j = \frac{\Delta^j y_0}{j!\,(\Delta x)^j}. \qquad (5.9)$$

Это выражение и выражение (5.8) позволяет решить задачу интерполяции.

Программы 5.2 реализуют этот метод интерполяции при $N=2$:

Программа 5.2/21

Р2	С/П	Р3	↑	F2	—	Р4	С/П	↑	F2	+	Р5
С/П	↑	С/П	—	Р6	↑	С/П	Р7	—	↑	F6	×
Р8	F3	↑	2	×	/—/	↑	F5	+	↑	F8	×
↑	F7	Fx^2	÷	2	÷	↑	F2	+	Р2	F4	↑
F6	×	↑	F7	÷	↑	F2	+	С/П			

Программа 5.2/34

П0	С/П	П1	—	С/П	+	ИП1	—	С/П	П2
С/П	П3	—	П4	×	С/П	П5	ИП4	XY	—
×	ИП5	Fx^2	2	×	÷	ИП4	ИП5	÷	ИП1
ИП0	—	×	+	ИП0	+	С/П			

Инструкция к программам 5.2: В/О; набрать y_0 С/П; y_1 С/П; y_2 С/П; x_1 С/П; x_0 С/П; набрать Δx С/П; результат $\hat{f}(x)$ содержится в РХ.

Известны также и другие методы интерполяции, например метод Ньютона для интерполяции назад, методы Гаусса, Бесселя и др. [13, 17, 41].

Отметим, что интерполяция полиномами является точной лишь в тех случаях, когда интерполируемая функция является полиномом той же степени. В других случаях полиномиальная интерполяция дает лишь некоторое приближение, т. е. решается задача аппроксимации функции полиномом.

Так как существует один и только один полином степени N, проходящий через $N+1$ точек, то все методы построения интерполяционных полиномов сводятся к получению одного и того же полинома (при фиксированном наборе точек $\{y_0,\ldots,y_N\}$). Поэтому выбор того или иного метода интерполяции в основном определяется различием в требованиях к емкости памяти микрокалькулятора и времени вычислений.

Добавим, что в некоторых таблицах элементарных и специальных функций (см., например, [26]) шаг дискретизации выбирается таким, чтобы при простейшей линейной интерполяции обеспечить четыре-пять верных знаков результата, т. е. обеспечить точность, достаточную во многих случаях. При этом в таблицах часто содержится информация о максимальной ошибке линейной интерполяции. Если требуется большая точность интерполяции, то следует использовать интерполяционную процедуру более высокого порядка. В таблицах иногда приводят информацию о числе точек и соответственно о порядке интерполяционного многочлена, обеспечивающих полную табличную точность [26].

5.2. Численное интегрирование и дифференцирование функций

Основу методов численного интегрирования составляют алгоритмы интерполяции, рассмотренные в § 5.1. Здесь рассматривается лишь одна из возможных задач интегрирования, когда значения подынтегральной функции $f(x)$ заданы лишь на конечном множестве точек x_i $(i=0,\dots,n)$ интервал $[a, b]$, причем $x_0=a$, $x_n=b$, $x_{i+1}-x_i=\Delta x$. Если функция $f(x)$ задана аналитически, то можно использовать программы 5.3. Необходимо найти приближенное значение к определенному интегралу

$$I=\int_a^b f(x)\,dx.$$

При сделанных допущениях известные методы численного интегрирования различаются лишь видом интерполяционного многочлена, используемого для приближения функции $f(x)$ по ее значениям $\{f(x_0),\dots$ $\dots, f(x_n)\}$, заданным в дискретных точках $\{x_0, \dots, x_n\}$.

Простейший алгоритм интегрирования (формула прямоугольников) получается при интерполяции функции полиномом нулевого порядка:

$$\hat{I}=\Delta x\,[f(x_0)+f(x_1)+\dots+f(x_{n-1})]. \tag{5.10}$$

При линейной интерполяции функции $f(x)$ получается так называемая формула трапеций, которая аппроксимирует интеграл суммой площадей трапеций с основанием $[x_i, x_{i+1}]$ и высотой, изменяющейся линейно от $f(x_i)$ до $f(x_{i+1})$. Формула трапеций имеет вид

$$\hat{I}=\frac{\Delta x}{2}\,[f(x_0)+2f(x_1)+2f(x_2)+\dots+2f(x_{n-1})+f(x_n)]. \tag{5.11}$$

Точность интегрирования можно повысить, если заменить линейную интерполяцию квадратичной. При этом получается формула, известная как правило Симпсона:

$$\hat{I}=\frac{1}{3}\,\Delta x\,[f(x_0)+4f(x_1)+2f(x_2)+4f(x_3)+\dots+4f(x_{n-1})+$$
$$+f(x_n)]. \tag{5.12}$$

Этот метод реализуется в программах 5.3:

Программа 5.3/21

↑	F8	PX=0	F4	XY	P8	С/П	↑	4	×	↑	F8
+	P8	С/П	↑	2	×	↑	F8	+	P8	С/П	БП
F1	+	↑	F2	×	3	÷	P7	С/П			

Программа 5.3/34

↑	С/П	↑	4	×	+	С/П	↑	2	×
+	БП	01	+	С/П	×	3	÷	С/П	

Инструкция к программе 5.3/21: В/О; набрать 0 P8; Δx P2; поочередно ввести значения $f(x_i)$ С/П $(i=1,\dots$ $\dots, n-1)$; набрать последнее значение $f(x_n)$ В/О С/П; результат $\hat{I}$ содержится в PX и P7.

Инструкция к программе 5.3/34: В/О; поочередно ввести значения $f(x_i)$ С/П $(i=1,\dots,n-1)$; набрать последнее значение $f(x_n)$ БП 13 С/П; набрать Δx С/П; результат $\hat{I}$ содержится в PX.

Для повышения точности интегрирования иногда используют интерполяционные многочлены и более высоких порядков k. В этом случае формула для оценки интеграла имеет вид

$$\hat{I}=C_k\,\Delta x\sum_{j=0}^{n-1/k-1}\sum_{i=0}^{k} w_i\,f(x_{i+j+k}), \tag{5.13}$$

где C_k и w_i — некоторые коэффициенты, значения которых для различных $k\leqslant 7$ приведены в табл. 5.1 [41].

Здесь число отсчетов дискретизованной функции должно удовлетворять условию $n=mk+1$, где m — целое число.

Например, для $k=3$ из выражения (5.13) и данных табл. 5.1 получается следующая формула:

$$\hat{I}=\frac{3}{8}\,\Delta x\,[f(x_0)+3f(x_1)+3f(x_2)+2f(x_3)+3f(x_4)+$$
$$+3f(x_5)+2f(x_6)+\dots+3f(x_{n-1})+f(x_n)]. \tag{5.14}$$

Таблица 5.1

k	C_k	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7
2	1/3	1	4	1	—	—	—	—	—
3	3/8	1	3	3	1	—	—	—	—
4	2/45	7	32	12	32	7	—	—	—
5	5/288	19	75	50	50	75	19	—	—
6	1/140	41	216	27	272	27	216	41	—
7	7/17280	751	3577	1323	2989	2989	1323	3577	751

Шаг интегрирования и порядок интерполирующего многочлена определяются требуемой точностью и допустимым временем решения задачи, а также видом подынтегральной функции. Обычно они подбираются методом последовательных приближений, сравнивая результаты интегрирования, полученные до и после увеличения значений n или k.

Часто требуется определить приближенно производную функции, заданной в дискретных точках:

$$y_i = f(x_i) \quad (i = 0, \ldots, n).$$

Иногда для оценки производной используют простую и очевидную формулу [41]:

$$\hat{y}'_i = \frac{y_{i+1} - y_{i-1}}{2\Delta x} \quad (i = 1, \ldots, n-1). \qquad (5.15)$$

Оценим, насколько близка эта вычислительная операция к дифференцированию. Для этого рассмотрим функцию $y = e^{j\omega x}$.

В результате точного дифференцирования

$$y'_i = y'(x)\big|_{x=i\Delta x} = j\omega e^{j\omega\Delta x i}. \qquad (5.16)$$

В то же время выражение (5.15) дает оценку производной:

$$\hat{y}'_i = e^{j\omega\Delta x i}\,\frac{e^{j\omega\Delta x} - e^{-j\omega\Delta x}}{2\Delta x} = j\omega e^{j\omega\Delta x i}\,\frac{\sin\omega\Delta x}{\omega\Delta x}. \qquad (5.17)$$

Степень соответствия оценки истинному значению производной характеризуется отношением выражений (5.17) и (5.16); $\sin\omega\Delta x/\omega\Delta x$. Таким образом, алгоритм (5.15) существенно искажает значения производной для быстроизменяющихся функций. Поэтому на практике

прибегают к более сложным алгоритмам. Они основываются на рассмотренных выше формулах интерполяции.

Поясним это на примере.

Пусть интерполирующий многочлен имеет вид [см. выражение (5.6)]:

$$P_2(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1),$$

где

$$C_0 = y_0; \quad C_1 = (y_1 - y_0)/\Delta x; \quad C_2 = (y_2 - y_1 + y_0)/2(\Delta x)^2.$$

Продифференцируем его и получим

$$P'_2(x) = C_1 + C_2(2x - x_0 - x_1).$$

При $x = x_0$ получается следующее значение производной полинома, принимаемой в качестве оценки производной функции:

$$\hat{y}'_0 \equiv P'_2(x_0) = (-3y_0 + 4y_1 - y_2)/2\Delta x. \qquad (5.18)$$

Эта величина аппроксимирует производную по трем точкам с использованием правых разностей. Аналогичным образом можно получить выражения для аппроксимации производной по трем точкам с использованием центральных и левых разностей. Можно также повысить степень аппроксимирующего полинома и тем самым увеличить число точек, по которым оценивается производная. Некоторые алгоритмы определения правой производной приведены в табл. 5.2.

Программы 5.4 предназначены для численного дифференцирования функций по пяти точкам с использованием центральных разностей:

Программа 5.4/21

Р3	F2	1	2	×	Р2	F7	↑	F3	—	Р8	F4
↑	F6	—	8	×	↑	F8	+	↑	F2	÷	Р8
F6	Р7	F5	Р6	F4	Р5	F3	Р4	F8	С/П	Р3	БП
Р1											

Программа 5.4/34

П2	0	П4	П5	П6	П7	С/П	П3	ИП2	1
2	×	П2	ИП7	ИП3	—	П8	ИП4	ИП6	—
8	×	ИП8	+	ИП2	÷	П8	ИП6	П7	ИП5
П6	ИП4	П5	ИП3	П4	ИП8	С/П	П3	БП	13

Таблица 5.2

Тип аппроксимации	Число точек аппроксимации		
	3	5	7
С использованием правых разностей	$\hat{y}'_i = \frac{1}{2\Delta x}\left(-y_{i+2} + 4y_{i+1} - 3y_i\right)$	$\hat{y}'_i = \frac{1}{12\Delta x}\left(-3y_{i+4}\ 16y_{i+3} - 36y_{i+2} + 48_{i+1} - 25y_i\right)$	—
С использованием центральных разностей	$\hat{y}'_i = \frac{1}{2\Delta x}\left(y_{i+1} - y_{i-1}\right)$	$\hat{y}'_i = \frac{1}{12\Delta x}\left(-y_{i+2} + 8y_{i+1} - 8y_{i-1} + y_{i-2}\right)$	$\hat{y}'_i = \frac{1}{60\Delta x}\left(y_{i+3} - 9y_{i+2} + 45y_{i+1} - 45y_{i-1} + 9y_{i-2} - y_{i-3}\right)$
С использованием левых разностей	$\hat{y}'_i\ \frac{1}{2\Delta x}\left(3y_i - 4y_{i-1} + y_{i-2}\right)$	$\hat{y}'_i = \frac{1}{12\Delta x}\left(25y_i - 48_{i-1}\right) + 36y_{i-2} - 16y_{i-3} + 3y_{i-4})$	—

Таблица 5.3

Тип аппроксимации	Число точек аппроксимации		
	3	5	7
С использованием правых разностей	$\hat{y}''_i = \frac{1}{(\Delta x)^2}\left(y_{i+2} - 2y_{i+1} + y_i\right)$	$\hat{y}''_i - \frac{1}{12(\Delta x)^2}\left(11y_{i+4} - 56y_{i+3} + 114y_{i+2} - 104y_{i+1} + 35y_i\right)$	—
С использованием центральных разностей	$\hat{y}''_i = \frac{1}{(\Delta x^2)}\left(y_{i+1} - 2y_i + y_{i-1}\right)$	$\hat{y}''_i = \frac{1}{12(\Delta x)^2}\left(-y_{i+2} - 16y_{i+1} - 30y_i + 16y_{i-1} - y_{i-2}\right)$	$\hat{y}''_i = \frac{1}{180(\Delta x)^2}\left(2y_{i+3} - 27y_{i+2} + 270y_{i+1} - 490y_i + 270y_{i-1} - 27y_{i-2} + 2y_{i-3}\right)$
С использованием левых разностей	$\hat{y}''_i = \frac{1}{(\Delta x)^2}\left(y_i - 2y_{i-1} + y_{i-2}\right)$	$\hat{y}''_i = \frac{1}{12(\Delta x)^2}\left(35y_i - 104y_{i-1} + 104y_{i-2} - 56_{i-3} + 11y_{i-4}\right)$	—

Эти программы формируют оценку значения производной с отставанием на два отсчета: после ввода значения функции y_i ($i \geqslant 5$) получается оценка $\hat{y}'_{i-2}$.

Инструкция к программе 5.4/21: В/О; набрать 0 Р4 Р5 Р6 Р7; набрать Δx Р2; поочередно набрать y_i С/П; начиная с $i=5$ в РХ и Р8 содержится $\hat{y}'_{i-2}$.

Инструкция к программе 5.4/34: В/О; набрать Δx С/П; поочередно набирать y_i С/П; начиная с $i=5$ в РХ и Р8 содержится $\hat{y}'_{i-2}$.

Аналогично могут быть получены алгоритмы для оценки производных более высоких порядков [43]. В табл. 5.3 и 5.4 приводятся выражения для вторых и третьих производных [41].

Таблица 5.4

Тип аппроксимации	Число точек аппроксимации	
	5	7
С использованием правых разностей	$\hat{y}'''_i = \frac{1}{2(\Delta x)^3}\left(-3y_{i+4} + 14y_{i+3} - 24y_{i+2} + 18_{i+1} - 5y_i\right)$	—
С использованием центральных разностей	$\hat{y}'''_i = \frac{1}{2(\Delta x)^3}\left(y_{i+2} - 2y_{i+1} + 2y_{i-1} - y_{i-2}\right)$	$\hat{y}'''_i = \frac{1}{8(\Delta x)^3}\left(-y_{i+3} + 8y_{i+2} - 13y_{i+1} + 13y_{i-1} - 8_{i-2} + y_{i-3}\right)$
С использованием левых разностей	$\hat{y}'''_i = \frac{1}{2(\Delta x)^3}\left(5y_i - 18y_{i-1} + 24y_{i-2} - 14_{i-3} + 3y_{i-4}\right)$	—

Решение алгебраических уравнений. Общий вид алгебраического уравнения n-й степени имеет вид

$$a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 = 0. \qquad (5.19)$$

Известно много методов решения алгебраических уравнений. Некоторые из них принадлежат к числу старейших алгоритмов численного анализа. К ним относятся, например, методы Горнера, Греффе, Бернулли, Ратисхаузера, Ламера, Берстоу и др. [13, 17, 21, 42].

По сравнению с трансцендентными уравнениями (см. ниже) алгебраические уравнения имеют то преимущество, что заранее известно число их корней и, следовательно, известно, когда нужно остановить процесс решения.

Напомним некоторые полезные свойства алгебраических уравнений.

1. Алгебраическое уравнение n-й степени имеет n корней, которые могут быть действительными или комплексными.

2. Если все коэффициенты a_i ($i=0, \ldots, n$) действительны, то комплексные корни образуют комплексно-сопряженные пары.

3. Число положительных действительных корней равно или меньше (на целое число) числа перемен знаков в последовательности коэффициента a_i.

4. Число отрицательных действительных корней равно или меньше (на целое число) числа перемен знаков в последовательности коэффициентов a_i при замене x на $-x$.

5. Модули корней нормированного ($a_n=1$) уравнения не превышают более чем на единицу модуль наибольшего коэффициента a_m.

Рассмотрим методы решения алгебраических уравнений с действительными коэффициентами. Известно, что аналитические выражения существуют лишь для корней уравнений степени не выше четвертой [13].

В частности, для корней квадратного уравнения

$$a_2 x^2 + a_1 x + a_0 = 0$$

справедливы выражения

$$\hat{x}_{1,2} = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2 a_0}}{2a_2}.$$

При выполнении расчетов на микрокалькуляторах для уменьшения числа операций эту формулу целесообразно преобразовать:

$$\hat{x}_{1,2} = \frac{a_1}{2a_2} \pm \sqrt{\left(\frac{a_1}{2a_2}\right)^2 - \frac{a_0}{a_2}}\,. \tag{5.20}$$

Если подкоренное выражение отрицательно, то корни уравнения являются комплексными:

$$\hat{x}_{1,2} = \frac{a_1}{2a_2} \pm j\sqrt{\left|\left(\frac{a_1}{2a_2}\right)^2 - \frac{a_0}{a_2}\right|}. \tag{5.21}$$

Выражения (5.20) и (5.21) использованы в программах:

Программа 5.5/21

F3	2	÷	↑	F2	÷	/—/	P5	F4	↑	F2	÷
XY	F5	Fx^2	—	P6	PX<0	F/—/	/—/	F$\sqrt{}$	P6	↑	F5
+	P7	F5	↑	F6	—	P8	7	8	С/П	F$\sqrt{}$	P6
Fx^2	↑	F5	Fx^2	+	F$\sqrt{}$	P7	F6	↑	F5	÷	P8
5	6	7	8	С/П							

Программа 5.5/34

П0	С/П	П1	С/П	П2	÷	2	÷	/—/	П5
Fx^2	ИП0	ИП2	÷	—	FX⩾0	29	F$\sqrt{}$	П3	ИП5
+	П7	ИП5	ИП3	—	П8	7	8	С/П	/—/
F$\sqrt{}$	П6	Fx^2	ИП5	Fx^2	+	F$\sqrt{}$	П7	ИП6	ИП5
÷	П8	5	6	7	8	С/П			

В случае когда корни комплексные, в программах вычисляется также модуль и тангенс аргумента корня, лежащего в верхней полуплоскости.

Инструкция к программе 5.5/21: В/О; набрать a_0 Р4; a_1 Р3; a_2 Р2 С/П; после окончания вычислений, если содержимое регистра РХ <РХ>=78, то действительные корни $\hat{x}_1$ и $\hat{x}_2$ содержатся в Р7 и Р8; если <РХ>=5678, то действительная часть комплексных корней $\hat{x}_1$ и $\hat{x}_2$ содержится в Р5, модуль мнимой части — в Р6, модуль корней $|\hat{x}_1| = |\hat{x}_2|$ — в Р7, тангенс аргумента $\mathrm{tg}\{\arg\{\hat{x}_1\}\}$ — в Р8.

Инструкция к программе 5.5/34: В/О; набрать a_0 С/П; a_1 С/П; a_2 С/П. Размещение результатов в регистрах то же, что и в предыдущей программе.

Контрольный пример. Исходные данные: $a_2=6$, $a_1=5$, $a_0=4$. Результаты: $\hat{x}_{1,2}=-0{,}4166666\pm j0{,}7021791$, $|\hat{x}_{1,2}|=0{,}8164965$, $\mathrm{tg}\{\arg\{\hat{x}_1\}\}=-1{,}685229$.

Для решения кубического уравнения можно использовать метод Кардано [28]. В этом случае исходное кубическое уравнение

$$x^3 + a_2x^2 + a_1x + a_0 = 0 \tag{5.22}$$

подстановкой $x=y-(a_2/3)$ приводится к «неполному» виду:

$$y^3 + py + q = 0, \tag{5.23}$$

где

$$p = -\frac{a_2}{3} + a_1; \quad q = 2\left(\frac{a_2}{3}\right)^3 - \frac{a_2a_1}{3} + a_0.$$

Корни «неполного» кубического уравнения (5.23) равны

$$\hat{y}_1 = A + B; \quad \hat{y}_{2,3} = -\frac{A+B}{2} \pm j\frac{A-B}{2}\sqrt{3}, \tag{5.24}$$

где

$$A = \sqrt[3]{-\frac{q}{2} + \sqrt{Q}}; \quad B = \sqrt[3]{-\frac{q}{2} - \sqrt{Q}};$$

$$Q = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2;$$

причем в качестве A и B берутся любые значения кубических корней из соответствующих комплексных чисел, удовлетворяющих соотношению $AB=-p/3$.

Возможно использование комбинированного метода, в котором значение действительного корня (всегда имеется хотя бы один) приближенно находится одним из численных итерационных методов, рассмотренных ранее. После этого порядок уравнения можно понизить на единицу и для решения квадратного уравнения использовать аналитические выражения. Программы 5.6 реализуют такой комбинированный метод:

Программа 5.6(1)/21

F4	Р6	F6	2	×	Р6	/—/	ПП	Р6	РХ<0	Р↑	F6
2	÷	Р6	↑	F5	XY	+	Р5	—	РХ=0	F4	F5
С/П	ПП	F6	РХ⩾0	XY	F5	↑	F6	—	Р5	БП	XY
Р5	F5	↑	F2	ПП	Р—	F3	ПП	Р—	F4	XY	+
×	В/О	+	↑	F5	×	↑	В/О				

Программа 5.6(2)/21

F2	↑	F5	+	Р8	↑	F5	×	↑	F3	+	Р7
↑	F5	×	↑	F4	+	Р→	F8	2	÷	/—/	Р6
Fx^2	↑	F7	—	РХ<0	РВП	/—/	F√	Р8	6	0	8
0	С/П	F√	↑	F6	XY	Р6	—	Р7	F6	+	Р6
6	7	С/П									

Программа 5.6/34

П0	С/П	П1	С/П	П2	П4	ИП4	2	×	П4
/—/	ПП	38	FX<0	06	ИП4	2	÷	П4	ИП3
XY	+	П3	—	FX=0	28	БП	55	ПП	39
FX⩾0	15	ИП3	ИП4	—	П3	БП	15	П3	ИП3
ИП0	ПП	51	ИП1	ПП	51	ИП2	+	ИП2	×
В/О	+	ИП3	×	В/О	ИП3	ИП0	+	П6	ИП3
×	ИП1	+	П5	ИП3	×	ИП2	+	П7	ИП6
2	÷	/—/	П4	Fx^2	ИП5	—	FX<0	85	/—/
F√	П6	4	6	С/П	F√	П8	ИП4	+	П4
FBx	ИП8	—	П5	4	5	С/П			

Инструкция к программе 5.6/21: ввести программу 5.6(1)/21; В/О; набрать a_2 Р2; a_1 Р3; a_0 Р4; С/П; после окончания вычислений вещественный корень $\hat{x}_1$ содержится в РХ и Р5; не изменяя содержимого регистров памяти, ввести программу 5.6(2)/21; В/О; С/П; после окончания вычислений, если <РХ>=67, то вещественные корни $\hat{x}_2$ и $\hat{x}_3$ содержатся в Р6 и Р7; если <РХ>= =6080, то вещественная часть комплексных корней содержится в Р6, а модуль мнимой — в Р8.

Инструкция в программе 5.6/34: В/О; набрать a_2 С/П; a_1 С/П; a_0 С/П; если после окончания вычислений <РХ>=45, то вещественные корни $\hat{x}_1$, $\hat{x}_2$ и $\hat{x}_3$ содержатся в Р3, Р4 и Р5; если <РХ>=46, то вещественный корень $\hat{x}_1$ содержится в Р3, вещественная часть корней $\hat{x}_2$ и $\hat{x}_3$ — в Р4, модуль мнимой части — в Р6.

Контрольный пример. Исходные данные: $a_2=-6$, $a_1=11$, $a_0=-6$. Результаты: $\hat{x}_1=1$, $\hat{x}_2=3$; $\hat{x}_3=2$.

Время вычисления корней зависит от конкретного вида уравнения. Для контрольного примера оно составляет около 720 с.

Более подробно методы решения алгебраических уравнений высоких степеней описаны в работах [13, 21, 41], а в работе [28] приводятся дополнительные программы для микрокалькуляторов.

Анализ устойчивости алгебраических многочленов. Известно [23], что определение устойчивости линейных динамических систем сводится к выяснению расположения на комплексной частотной плоскости корней так называемого характеристического полинома системы

$$A(p) = a_n p^n + a_{n-1} p^{n-1} + \ldots + a_0, \qquad (5.25)$$

представляющего собой знаменатель передаточной функции системы. Считается, что коэффициенты a_i многочлена (5.25) вещественны, а старший коэффициент $a_n>0$.

Многочлен (5.25), все корни которого лежат в левой полуплоскости комплексной переменной p, соответствует устойчивой динамической системе и поэтому называется устойчивым многочленом. Можно ли, не вычисляя корней полинома (что для многочленов степени, превышающей $n=3\div4$, весьма сложно), сделать вывод об его устойчивости? Ответ дают так называемые критерии устойчивости — теоремы, которые определяют условия, накладываемые на коэффициенты многочлена a_i, при выполнении которых его корни лежат в требуемой области. Приведем некоторые из них [23].

1. Многочлен первой или второй степени устойчив тогда и только тогда, когда все его коэффициенты положительны.

2. Теорема Стодолы. Если многочлен устойчив, то все его коэффициенты положительны. Обратное утверждение неверно уже для многочлена третьей степени.

3. Критерий Вышнеградского. Для устойчивости многочлена третьей степени необходимо и достаточно, чтобы выполнялись условия:

$$a_i > 0 \quad (i = 0, \ldots, 3); \qquad a_1 a_2 > a_0 a_3.$$

Видно, что анализ устойчивости многочленов первого—третьего порядка весьма прост и не требует разработки специальных вычислительных программ. Сложнее дело обстоит с многочленами более высокого порядка.

4. Рассмотрим один из широко используемых критериев устойчивости многочленов — критерий Гурвица. Предварительно дадим основные определения.

Матрицей Гурвица многочлена (5.25) называется квадратная матрица размера $n\times n$, имеющая вид

$$\begin{bmatrix} a_{n-1} & a_n & 0 & 0 & \ldots & 0 \\ a_{n-3} & a_{n-2} & a_{n-1} & a_n & \ldots & 0 \\ a_{n-5} & a_{n-4} & a_{n-3} & a_{n-2} & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \ldots & a_0 \end{bmatrix}. \qquad (5.26)$$

Определителями Гурвица называются главные миноры Δ_i ($i=0, ..., n$) матрицы (5.26). Иными словами, они представляют собой определители квадратных матриц, «вырезаемых» из матрицы Гурвица (5.26), начиная с ее левого верхнего угла:

$$\Delta_1 = a_{n-1};$$

$$\Delta_2 = \begin{vmatrix} a_{n-1} & a_n \\ a_{n-3} & a_{n-2} \end{vmatrix};$$

$$\Delta_3 = \begin{vmatrix} a_{n-1} & a_n & 0 \\ a_{n-3} & a_{n-2} & a_{n-1} \\ a_{n-5} & a_{n-4} & a_{n-3} \end{vmatrix};$$

$$\cdots\cdots\cdots\cdots$$

$$\Delta_n = a_0 \Delta_{n-1}.$$

Теорема Гурвица. Для устойчивости многочлена n-й степени необходимо и достаточно, чтобы все определители Гурвица были положительны: $\Delta_i>0$ ($i=0, ..., n$).

Непосредственное вычисление определителей Гурвица используют, как правило, лишь при $n\leqslant 6$. Программы 5.7 и 5.8 предназначены для анализа устойчивости многочленов четвертого и пятого порядка соответственно.

Программа 5.7/21

F3	↑	F4	×	P7	F2	↑	F5	×	↑	F7	—
P7	PX≥0	×	0	С/П	F6	↑	F3	Fx^2	×	↑	F5
÷	↑	F7	+	PX<0	2	F6	/—/	PX<0	2	1	С/П

Программа 5.7/34

П0	СП	П1	С/П	П2	С/П	П3	×	СП	ИП1
×	XY	—	П5	FX≥0	18	0	С/П	ИП0	ИП3
Fx^2	×	ИП1	÷	ИП5	+	FX<0	16	ИП0	/—/
FX<0	16	1	С/П						

Программа 5.8/21

F4	↑	F3	×	P8	F5	↑	F2	×	↑	P→	F8
—	PX≥0	×	0	С/П	P→	F7	↑	F2	×	P→	F6
↑	F3	×	↑	P←	—	↑	F3	×	↑	F5	÷
↑	P←	+	PX<0	2	F7	↑	F4	×	↑	P←	×
P←	F6	↑	F5	×	↑	F8	×	↑	P→	—	С/П

Программа 5.8/34

П0	С/П	П1	С/П	П2	×	С/П	П3	С/П	П4
×	×	П6	FBx	С/П	П5	ИП2	×	П7	XY
—	FX≥0	25	0	С/П	ИП1	ИП4	×	ИП0	ИП5
×	—	ИП4	×	ИП2	÷	+	FX<0	23	ИП6
ИП7	ИП0	×	ИП3	×	—	С/П			

Инструкция к программе 5.7/21: В/О; набрать a_0 Р6; a_1 Р5; a_2 Р4; a_3 Р3; a_4 Р2; С/П; если <РХ>=0, то полином неустойчив, если <РХ>=1 — полином устойчив.

Инструкция к программе 5.7/34: В/О; набрать a_0 С/П; a_1 С/П; a_2 С/П; a_3 С/П; a_4 С/П, если <РХ>=0, то полином неустойчив, если <РХ>=1, то устойчив.

Инструкция к программе 5.8/21: В/О; набрать a_0 Р7; a_1 Р6; a_2 Р5; a_3 Р4; a_4 Р3: a_5 Р2; С/П; если <РХ> >0, то полином устойчив, если <РХ>⩽0, то неустойчив.

Инструкция к программе 5.8/34: В/О; набрать a_0 С/П; a_1 С/П; a_2 С/П; a_3 С/П; a_4 С/П; a_5 С/П; если <РХ>>0, то полином устойчив, если <РХ>⩽0, то неустойчив.

При $n\geqslant 6$ целесообразно использовать другую форму критерия Гурвица, когда оценивается знак коэффициентов разложения в цепную дробь отношения четной и нечетной частей исследуемого полинома. Этот метод рассмотрен в [28, 31].

Численное решение уравнений. Рассмотрим методы численного определения вещественных корней нелинейных (алгебраических и трансцендентных) уравнений.

Пусть $y=f(x)$—полином или трансцендентная функция одной действительной переменной x. Задача состоит в том, чтобы найти корень или корни уравнения

$$f(x) = 0. \tag{5.27}$$

Начнем с рассмотрения метода половинного деления. Этот метод прост и, что особенно важно, обеспечивает любую наперед заданную точность решения, которая ограничивается лишь наименьшим возможным числом с плавающей точкой в используемом микрокалькуляторе.

Решение производится следующим образом. Сначала вычисляются значения функции $f(x_i)$ в дискретных точках из области определения функции. Это делается до тех пор, пока не будут найдены два последовательных значения $f(x_n)$ и $f(x_{n+1})$, имеющие противоположные знаки. Искомый корень лежит в интервале между x_n и x_{n+1}. Нулевое приближение $\hat{x}^{(0)}$ к корню уравнения определяется как среднее значений x_n и x_{n+1}:

$$\hat{x}^{(0)} = \frac{x_{n+1} + x_n}{2}. \tag{5.28}$$

Далее вычисляется значение функции $f(\hat{x}^{(0)})$. Если ее знак совпадает со знаком $f(x_n)$, то в дальнейшем принимают $x_n=\hat{x}^{(0)}$. Если $f(\hat{x}^{(0)})$ совпадает по знаку с $f(x_{n+1})$, то величиной $\hat{x}^{(0)}$ заменяют величину x_{n+1}. В результате интервал, в котором заключено истинное значение корня, сужается вдвое. Далее операция поиска повторяется.

Используется несколько критериев окончания итерационного процесса. В частности, процесс может заканчиваться, если выполняется условие

$$\left| f(\hat{x}^{(i)}) \right| \leqslant \delta, \tag{5.29}$$

где $\hat{x}^{(i)}$ — оценка корня уравнения, полученная на i-й итерации; δ — заданная положительная величина, характеризующая точность решения (невязка решения).

Другим распространенным критерием является условие

$$\left| \hat{x}^{(i)} - \hat{x}^{(i-1)} \right| \leqslant \delta, \tag{5.30}$$

которое определяет окончание вычислений, как только последовательные оценки корней перестанут существенно изменяться.

Для иллюстрации рассмотренного метода* может использоваться программа 5.9/21 вычисления функции $y = \arcsin a$ путем решения уравнения $\sin x - a = 0$ с использованием критерия (5.30):

Программа 5.9/21

Р2	1	Р5	Рπ	2	÷	Р7	/—/	Р6	ПП	F6	↑
F5	—	Fx^2	F√	↑	F3	—	РХ ≥ 0	F4	F8	Р5	БП
1	F8	↑	Рπ	÷	↑	1	8	0	×	↑	F8
С/П	F7	↑	F6	+	↑	2	÷	Р8	Psin	↑	F2
—	РХ ≥ 0	Р9	F8	Р7	В/О	F8	Р6	В/О			

Инструкция к программе 5.9/21: В/О; набрать δ Р3; a С/П; результат $\hat{x}$, выраженный в радианах, содержится в РХ и Р8, в градусах — РУ.

Программы 5.10 являются универсальными.

Программа 5.10/21

F3	ПП	F6	Р6	F3	↑	F2	+	2	÷	Р5	↑
F2	—	↑	F4	—	РХ ≥ 0	/—/	F5	ПП	F6	↑	F6
×	РХ ≥ 0	F5	F5	Р3	БП	F↑	F5	Р2	БП	F↑	F5
С/П											

* Рассмотренные в этом пункте программы вычисления обратных тригонометрических функций не только иллюстрируют рассматриваемые методы, но и расширяют возможности Б3-21, в котором отсутствуют встроенные обратные тригонометрические функции.

Программа 5.10/34

П0	С/П	П1	С/П	П2	ИП1	ПП	38	П3	ИП0
ИП1	+	2	÷	П4	ИП0	—	ИП2	—	FX ≥ 0
36	ИП4	ПП	38	ИП3	×	FX ≥ 0	32	ИП4	П1
БП	05	ИП4	П0	БП	05	ИП4	С/П		

Перед их выполнением необходимо занести в свободную зону программной памяти программу вычисления функции $y = f(x)$. Здесь также используется критерий (5.30).

Инструкция к программе 5.10/21: В/О; набрать x_n Р2; x_{n+1} Р3; δ Р4; С/П; результат $\hat{x}$ содержится в РХ и Р5. Подпрограмма вычисления функции $f(x)$ начинается с адреса 61 и может достигать 23 шагов, значение аргумента содержится в РХ, вычисленное значение функции также должно быть в РХ, свободные регистры Р7 и Р8.

Инструкция к программе 5.10/34: В/О; набрать x_n С/П; x_{n+1} С/П; δ С/П; результат $\hat{x}$ содержится в РХ и Р4. Подпрограмма вычисления функции $f(x)$ начинается с адреса 38 и может достигать 60 шагов, значение аргумента содержится в РХ, в РХ должно быть помещено также значение $f(x)$, свободные регистры Р5—РД.

Другим широко используемым методом решения уравнений является метод хорд, называемый часто также методом ложного положения. В его основе лежит линейная интерполяция функции по двум ее значениям $f(x_n)$ и $f(x_{n+1})$, имеющим противоположные знаки. Прямая, проведенная через эти точки, пересекает ось OX в точке

$$\hat{x}^{(0)} = x_n - f(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}. \tag{5.31}$$

Это значение оценки корня используется для определения значения $f(\hat{x}^{(0)})$, которое сравнивается с $f(x_n)$ и $f(x_{n+1})$ и в дальнейшем используется вместо того из них, с которым совпадает по знаку. Соответственно, $\hat{x}^{(0)}$ заменяет величину x_n или x_{n+1}. Далее итерационный процесс повторяется.

Достоинством метода хорд является более быстрая сходимость решения по сравнению с методом половинного деления.

Для иллюстрации возможностей этого метода приведем программу 5.11/21 вычисления функции $\arccos a$ путем решения уравнения $\cos x - a = 0$:

Программа 5.11/21

P2	1	+	/—/	P5	1	↑	F2	—	P4	0	P6
Pπ	P7	F5	↑	F4	—	↑	F4	÷	P8	F6	↑
F7	—	↑	F8	÷	↑	F6	+	P8	Pcos	↑	F2
—	PX⩾0	7	↑	P4	F8	P6	БП	F8	↑	P5	F8
P7	XY	Fx^2	F$\sqrt{}$	↑	F3	—	PX<0	P×	F8	С/П	

Здесь использован критерий (5.29).

Инструкция к программе 5.11/21: В/О; набрать δ Р3, a С/П; результат $\hat{x}$, выраженный в радианах, содержится в РХ и Р8, время вычислений около 30 с.

Наибольшую популярность при решении нелинейных уравнений приобрел метод Ньютона — Рафсона. Это вызвано высокой скоростью сходимости решения. В отличие от двух предыдущих методов здесь не требуется находить значения функции, имеющие противоположные знаки. Каждое новое приближение $\hat{x}^{(i)}$ к корню уравнения вычисляется как точка пересечения с осью OX касательной к функции $f(x)$ в точке $\hat{x}^{(i-1)}$:

$$\hat{x}^{(i)} = \hat{x}^{(i-1)} - f(\hat{x}^{(i-1)})/f'(\hat{x}^{(i-1)}). \tag{5.32}$$

Успех применения метода Ньютона — Рафсона во многом определяется удачным выбором начального приближения $\hat{x}^{(0)}$.

Программа 5.12/21 служит для вычисления функции $\operatorname{arctg} a$ путем решения уравнения $\operatorname{tg} x - a = 0$:

Программа 5.12/21

P2	0	P4	F4	↑	Psin	↑	F4	Pcos	P5	÷	P8
↑	F2	—	P7	Fx^2	F$\sqrt{}$	↑	F3	—	PX⩾0	F—	F7
↑	F5	Fx^2	×	/—/	↑	F4	+	P4	F6	↑	F4
+	PX<0	P7	+	БП	P↑	F6	↑	F4	—	PX<0	0
+	/—/	БП	P↑	F4	↑	F6	÷	9	0	×	С/П

Критерий окончания решения — выражение (5.29).

Инструкция к программе 5.12/21: В/О; набрать Рπ ↑ 2 ÷ Р6; δ Р3; a С/П; результат $\hat{x}$ (в градусах) содержится в РХ, $\hat{x}$ (в радианах) — в Р4; время вычислений около 40 с.

Другими известными методами решения нелинейных уравнений являются метод секущих, метод простых итераций и др. [10, 13, 17, 34, 41].

Численное решение дифференциальных уравнений. Дифференциальные уравнения — уравнения, содержащие искомые функции, их производные любых порядков и независимые переменные. Дифференциальное уравнение первого порядка имеет вид

$$F(x, y, y') = 0 \text{ или } \frac{dy}{dx} = f(x, y). \tag{5.33}$$

В теории обыкновенных дифференциальных уравнений изучаются также уравнения высших порядков, а именно уравнения вида: $F(x, y, y', y'', \ldots, y^{n-1}, y^n) = 0$ и системы уравнений.

Чтобы решить дифференциальное уравнение n-го порядка, т. е. найти функцию $y(x)$, удовлетворяющую ему, необходимо знать значения зависимой переменной и ее производных до $(n-1)$-го порядка включительно при некоторых значениях независимой переменной. Если эти дополнительные условия задаются для некоторого фиксированного значения независимой переменной, то такая задача называется задачей с начальными условиями или задачей Коши. Если условия задаются при двух или более значениях независимой переменной, то задача называется краевой.

Как правило, решить обыкновенное дифференциальное уравнение в конечном виде невозможно, поэтому для их решения широко применяются приближенные методы: метод конечных разностей, графическое разложение в ряды. Большое значение имеют качественные методы. Рассмотрим применение метода Рунге — Кутта для решения дифференциальных уравнений первого и второго порядка.

Пусть дано дифференциальное уравнение первого порядка (5.33) и начальное условие $y(x_0) = y_0$. При использовании одной из форм метода Рунге — Кутта второго порядка [17], называемой часто исправленным методом Эйлера, приближенное значение функции y_{n+1} в точке $x_{n+1}=x_0+(n+1)h=x_n+h$, где h — шаг интегрирования по переменной x, определяется выражением

$$y_{n+1} = y_n + \frac{[f(x_n, y_n) + f(x_{n+1}, y^*)]\,h}{2}; \qquad (5.34)$$

$$y^* = y_n + hf(x_n, y_n). \qquad (5.35)$$

Этот метод имеет простой геометрический смысл — каждое последующее значение искомой функции находится с помощью предыдущего путем линейной экстраполяции с использованием усредненного тангенса угла наклона экстраполирующей прямой для двух точек: (x_n, y_n) и (x_{n+1}, y^*).

При реализации этого метода на микрокалькуляторе выражение (5.34) удобнее представить в виде

$$y_{n+1} = \frac{[y_n + y^* + hf(x_{n+1}, y^*)]}{2}. \qquad (5.36)$$

Программы 5.13 предназначены для вычисления последовательности значений искомой функции для значений независимой переменной, взятых с постоянным шагом:

Программа 5.13/21

ПП	Р/—/	↑	F2	×	↑	F4	+	Р4	F3	↑	F2
+	Р3	ПП	Р/—/	↑	F2	×	↑	F4	+	↑	F5
+	↑	2	÷	Р5	С/П	БП	Р0				

Программа 5.13/34

П0	С/П	П1	С/П	П2	П3	ПП	30	ИП0	×
ИП2	+	П2	ИП1	ИП0	+	П1	ПП	30	ИП0
×	ИП2	+	ИП3	+	2	÷	С/П	БП	05

Поскольку методическая погрешность вычисления функции имеет порядок h^3 [17], то для повышения точности целесообразно уменьшать шаг дискретизации. Однако при этом увеличивается время вычислений, кроме того возрастают погрешности за счет ошибок округления. К сожалению, не существует формул для выбора шага интегрирования до начала решения задачи. Поэтому на практике задачу решают несколько раз, постепенно уменьшая шаг интегрирования h до тех пор, пока отклонения вычисленных значений функции для одних и тех же значений независимой переменной не станут пренебрежимо малыми с точки зрения пользователя.

Инструкция к программе 5.13/21: В/О; набрать h Р2; x_0 Р3; y_0 Р4 Р5; для вычисления очередного значения функции y_{n+1} $(n=0, \ldots)$ нажимается клавиша С/П, y_{n+1} содержится в Р5, x_{n+1} — в Р3, y^* — в Р4. Подпрограмма вычисления функции $f(x, y)$ начинается с адреса 52 и может содержать до 28 шагов, значение x считывается из Р3, y — из Р4, значение $f(x, y)$ помещается в РХ.

Инструкция к программе 5.13/34: В/О; набрать h С/П; x_0 С/П; y_0; для вычисления очередного значения y_{n+1} $(n=0, \ldots)$ нажимается клавиша С/П; y_{n+1} содержится в РХ, x_{n+1} — в Р1, y^* — в Р2. Подпрограмма вычисления функции начинается с адреса 30 и может содержать до 68 шагов, x считывается из Р1, y — из Р2, значение $f(x, y)$ должно помещаться в РХ, свободные регистры — Р4 — РД.

Методы Рунге — Кутта можно использовать для решения систем дифференциальных уравнений первого порядка и, следовательно, для решения дифференциальных уравнений более высоких порядков, поскольку любое дифференциальное уравнение n-го порядка можно свести к n дифференциальным уравнениям первого порядка.

В частности, дифференциальное уравнение второго порядка

$$\frac{d^2 y}{dx^2} = f(x, y, y') \qquad (5.37)$$

с начальными условиями

$$y(x_0) = y_0; \quad y'(x_0) = y_0' \qquad (5.38)$$

можно преобразовать к системе из двух дифференциальных уравнений:

$$\left.\begin{aligned} \frac{dy}{dx} &= z; \\ \frac{dz}{dx} &= f(x, y, z) \end{aligned}\right\} \qquad (5.39)$$

с начальными условиями

$$y(x_0) = y_0, \quad z(x_0) = y_0'.$$

При использовании исправленного метода Эйлера получаются следующие выражения для расчета значений искомой функции [41]:

$$\left.\begin{aligned} z^*_{n+1} &= z_n + hf(x_n, y_n, z_n); \\ y_{n+1} &= y_n + \frac{h(z_n + z^*_{n+1})}{2}; \\ z_{n+1} &= \frac{z_n + z^*_{n+1} + hf(x_{n+1}, y_{n+1}, z^*_{n+1})}{2}. \end{aligned}\right\} \tag{5.40}$$

Этот алгоритм реализован в программах 5.14:

Программа 5.14/21

ПП	РСx	↑	F2	×	↑	F6	+	Р5	↑	F6	+
↑	F2	×	2	÷	↑	F4	+	Р4	F3	↑	F2
+	Р3	ПП	РСx	↑	F2	×	↑	F5	+	↑	F6
+	2	÷	Р5	Р6	С/П	БП	Р0				

Программа 5.14/34

П0	С/П	П1	С/П	П2	С/П	П3	П4	ПП	41
ИП0	×	ИП4	+	П3	ИП4	+	ИП0	×	2
÷	ИП2	+	П2	ИП1	ИП0	+	П1	ПП	41
ИП0	×	ИП3	+	ИП4	+	2	÷	С/П	БП
06									

Инструкция к программе 5.14/21: В/О; набрать h Р2; x_0 Р3; y_0 Р4; y'_0 Р5 Р6; для расчета очередного значения y_{n+1} $(n=0, \ldots)$ нажимается клавиша С/П; y_{n+1} содержится в Р4; x_{n+1} — в Р3, y'_{n+1} — в РХ, Р5 и Р6. Подпрограмма вычисления функции $f(x, y, y')$ начинается с адреса 72 и может достигать 18 шагов, значение y содержится в Р4, y' — в Р5, x — в Р3, $f(x, y, y')$ помещается в РХ.

Инструкция к программе 5.14/34: В/О; набрать h С/П; x_0 С/П; y_0 С/П; y'_0; для расчета очередного значения y_{n+1} $(n=0, \ldots)$ нажимается клавиша С/П; y_{n+1} содержится в Р2, x_{n+1} — в Р1, y'_{n+1} — в РХ. Подпрограмма вычисления функции $f(x, y, y')$ начинается с адреса 41 и может достигать 57 шагов; значение x считывается из Р1, y — из Р2, y' — из РХ, значение $f(x, y, y')$ помещается в РХ, свободные регистры Р5—РД.

5.4. Спектральный анализ функций

Методы дискретного преобразования Фурье. Среди известных методов спектрального анализа, широко используемых в различных областях науки и техники, наибольшее распространение получило преобразование

Фурье, представляющее вещественную функцию времени $x(t)$ с помощью комплексной спектральной функции частоты $X(j\,2\pi f)$ как

$$X(j2\pi f) = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi ft}\, dt. \tag{5.41}$$

При использовании вычислительных методов может быть определена спектральная функция ограниченных во времени или периодических процессов $x(t)$. При этом выражение (5.41) приобретает форму дискретного преобразования Фурье (ДПФ):

$$X\left(jk\frac{2\pi}{N}\right) = \sum_{n=0}^{N} x(n)\, e^{-nk2\pi/N}, \tag{5.42}$$

где N — число дискретных отсчетов $x(n)$ на интервале его наблюдения (длительности или периода повторения); k — номер отсчета спектральной функции (дискретная частота). Необходимое число N определяется исходя из требуемой точности получения спектральной функции. По теореме Котельникова шаг дискретизации функции $x(t)$ по времени процесса $x(t)$ определяется максимальной частотой, учитываемой в составе его спектральной функции [35]. Заметим, что согласно (5.42) спектральная функция, получаемая с помощью ДПФ, периодична по k (по частоте) с периодом $K=N$, таким образом, число комплексных чисел, представляющих ее на одном периоде, равно числу N дискретных отсчетов процесса $x(t)$.

Чтобы вычислить спектральную функцию, ее обычно записывают в форме

$$X\left(jk\frac{2\pi}{N}\right) = A_k + jB_k, \tag{5.43}$$

где

$$A_k = \sum_{n=0}^{N-1} x(n) \cos(2\pi kn/N);$$

$$B_k = \sum_{n=0}^{N-1} x(n) \sin(2\pi kn/N).$$

Для интегральных преобразований типа (5.41) характерно значительное число необходимых вычислительных операций, быстро возрастающее с увеличением N. Действительно, для определения одного отсчета спектральной функции (5.42) согласно формуле (5.43) требуется вычислить $2N$ значений косинуса (синуса), выполнить $2N$ умножений и $2N$ сложений. Поскольку максимальное значение k равно N, общий объем вычислений ДПФ растет пропорционально N^2, что значительно ограничивает возможности спектрального анализа, проводимого численными методами.

Спектральный анализ значительно упрощается с помощью специальных алгоритмов быстрого преобразования Фурье (БПФ) [17]. При этом число операций сокращается в результате использования свойства периодичности коэффициентов в формулах (5.43), однако это возможно лишь при условии одновременного учета всех дискретных отсчетов процесса $x(n)$, которые, следовательно, должны быть введены в память ЭВМ. Необходимость ввода в память всех отсчетов ограничивает применение БПФ лишь очень небольшим числом отсчетов. В работе [31] приведены программы реализаций БПФ на Б3-21 и Б3-34 при $n=6$ и $n=8$.

При выполнении на МК прямого вычисления ДПФ согласно формулам (5.42), (5.43) значения спектральной функции $X(jk\,2\pi/N)$ определяются при фиксированном k путем последовательного ручного ввода значений $x(n)$. Число этих отсчетов N, таким образом, не связано с требованиями к емкости памяти МК и поэтому не ограничено. Однако при другом значении k все расчеты повторяются в полном объеме, что в соответствии с отмечавшимся общим свойством ДПФ значительно увеличивает объем работы (в том числе ручной) при увеличении N. Уменьшение же числа отсчетов приводит к заметному ухудшению точности вычисления спектральной функции, что может быть объяснено наличием в составе функций (5.41), (5.42) быстро изменяющегося множителя [8]. Заметим, что это свойство особенно сильно проявляется при спектральном анализе импульсных процессов, сочетающих в себе участки быстрого (фронт и спад импульса) и плавного (вершина) изменения функции $x(t)$.

Приведем примеры программ прямого вычисления ДПФ [28]. Вычисление отсчетов спектральной функции $X(k\,2\pi/N)$ выполняется по программам 5.15:

Программа 5.15/21

P2	Cx	P5	P7	P8	Pπ	2	×	↑	F3	÷	↑
F4	×	P6	F5	↑	F6	×	Pe^{ix}	P←	F2	XY	×
P→	×	↑	F7	+	P7	P←	↑	F8	+	P8	F5
1	+	P5	↑	F3	—	PX≠0	F8	XY	С/П	P2	БП
2	F7	Fx^2	↑	F8	Fx^2	+	F√	С/П	БП	P0	

Программа 5.15/34

П2	Cx	П5	П7	П8	Fπ	2	×	ИП3	÷
ИП4	×	П6	ИП5	ИП6	×	↑	Fcos	ИП2	×
ИП7	+	П7	XY	Fsin	ИП2	×	ИП8	+	П8
КИП5	ИП3	ИП5	—	FX≠0	41	FBx	С/П	П2	БП
13	ИП7	Fx^2	ИП8	Fx^2	+	F√	С/П	БП	00

Отметим наличие в этих программах вспомогательных операций: после обработки каждого отсчета в РХ высвечивается требуемый номер следующего вводимого отсчета, что представляет значительное удобство и способствует сокращению числа возможных ошибок при большом числе ручных операций.

Инструкция к программе 5.15: В/О; набрать N Р3 (П3); k Р4 (П4); поочередно вводятся отсчеты x_i и нажимается клавиша С/П; после ввода всех временных отсчетов отсчет модуля спектральной функции помещается в РХ, действительная часть содержится в Р7, мнимая — в Р8 (выше в скобках приведены команды для Б3-34).

Другие известные алгоритмы и программы вычисления ДПФ на МК отличаются использованием различных способов уменьшения числа операций или повышения точности вычислений. Например, алгоритм Герцеля [28] предполагает предварительное вычисление множителей $\sin(2\pi k_i/N)$; $\cos(2\pi k_i/N)$ для нескольких значений k_i с последующим использованием их при вычислении ДПФ. В условиях ограниченных возможностей Б3-21 такой алгоритм реализуется в виде пакета из двух программ [31] с передачей данных из одной программы в другую, как рассматривалось в § 4.1. При этом обеспечивается одновременное вычисление трех отсчетов спектральной функции (программа 5.16/21):

Программа 5.16(1)/21

↑	Pπ	XY	÷	P2	F3	ПП	P→	P6	P←	P3	F4
ПП	P→	P7	P←	P4	F5	ПП	P→	P8	Cx	P2	P←
P5	С/П	0	P→	Cx	P→	F2	×	Pe^{ix}	XY	P→	F1
2	×	В/О									

Программа 5.16(2)/21

ПП	7	F6	ПП	Р6	F7	ПП	Р6	F8	ПП	Р6	С/П
ПП	7	F3	×	Р3	F6	ПП	5	F4	×	Р4	F7
ПП	5	F5	×	Р5	F8	ПП	5	С/П	×	2	F1/x
×	Р→	XY	Р←	+	Р→	Р→	↑	В/О	↑	F2	+
Р2	Р←	—	Р←	Р←	—	Р←	Р←	—	Р←	↑	В/О

Инструкция к программе 5.16/21: ввести первую часть программы: В/О; набрать k_1 Р3; k_2 Р4, k_3 Р5, $N/2$ С/П; в результате выполнения значение $\sin(2\pi k_1/N)$ содержится в Р3, $\sin(2\pi k_2/N)$ — в Р4, $\sin(2\pi k_3/N)$ — в Р5, $2\cos(2\pi k_1/N)$ — в Р6, $2\cos(2\pi k_2/N)$ — в Р7, $2\cos(2\pi k_3/N)$ — в Р8; не изменяя содержимого регистров, ввести вторую часть программы; В/О; поочередно вводить значения x_i в порядке убывания номеров, нажимая клавишу С/П; после ввода последнего отсчета A_0 содержится в Р2, B_{k_1} — в Р3, B_{k_2} — в Р4, B_{k_3} — в Р5, в регистрах кольцевого стека: в Р2 содержится A_{k_3}, в Р4 — A_{k_2}, в Р6 — A_{k_1}.

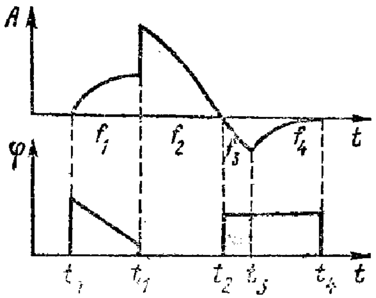


Рис. 5.1. Огибающая и фаза радиоимпульса

Возможности Б3-34 обеспечивают одновременный расчет пяти смежных значений спектральной функции [31].

Из других методов улучшения эффективности спектрального анализа отметим способ аппроксимации отрезков процесса $x(t)$ на отдельных шагах интегрирования [8], позволяющий получать соотношения для расчета ДПФ с улучшенной точностью. Такой подход можно рассматривать как пример реализации общей идеи аппроксимации процесса $x(t)$ функциями, допускающими интегрирование составляющих (5.41) в аналитическом виде, что ведет к резкому сокращению необходимого объема вычислений.

Определение спектральной функции при использовании кусочно-гладкой аппроксимации. Требования к объему вычислений значительно возрастают при спектральном анализе радиоимпульсов, т. е. отрезков быстроосциллирующих процессов. В таких случаях обычно рассматривают комплексную огибающую [35], представляющую собой относительно медленные функции изменения амплитуды и фазы радиоимпульса $A(t)$ и $\varphi(t)$ (рис. 5.1).

Представим комплексную огибающую сигнала $\dot f(t)=A(t)e^{j\varphi(t)}$ суперпозицией отрезков непрерывно дифференцируемых комплексных функций (рис. 5.1):

$$\dot f(t)=\sum_{k=1}^{L}\dot f_k(t), \tag{5.44}$$

где $\dot f_k(t)$ — функция, финитная на интервале (t_{k-1}, t_k).

Определим спектр комплексной огибающей $\dot f(t)$:

$$\dot F(\omega)=\sum_{k=1}^{L}\dot F_k(\omega), \tag{5.45}$$

где

$$\dot F_k(\omega)=\dot F_k\ (\omega;\, t_k)-\dot F_k(\omega;\, t_{k-1});$$

$$\dot F_k(\omega;\, t)=\int \dot f_k(t)\, e^{-j\omega t}\, dt\ —$$

первообразные интегралов:

$$\dot F_0(\omega)=0; \quad \dot F_{L+1}(\omega)=0.$$

Сгруппируем в выражении (5.45) слагаемые с одинаковыми t_k и образуем спектры

$$\dot S_{k-1}(\omega;\, t_{k-1})=\dot F_{k-1}(\omega;\, t_{k-1})-\dot F_k(\omega;\, t_{k-1}). \tag{5.46}$$

По физическому смыслу $\dot S_k(\omega;\, t_k)$ представляет собой мгновенный спектр функции $\dot f(t)$ в особой точке $t=t_k$. Общий спектр комплексной огибающей определяется суммой

$$\dot F(\omega)=\sum_{k=1}^{L}\dot S_{k-1}(\omega;\, t_{k-1}).$$

Представим комплексную функцию $\dot f_k(t)$ рядом Тейлора в точке t_{k-1}:

$$\dot f_k(t)=\sum_{i=0}^{\infty}\frac{\dot f_k^{(i)}(t_{k-1})}{i!}(t-t_{k-1})^i\,[1(t-t_{k-1})-1(t-t_k)],$$

где $1(t)$ — ступенчатая функция Хевисайда.

С учетом разрывности функции $\dot f_k(t)$ в этой точке находим спектр

$$\dot F_k(\omega;\, t_{k-1})=-\left[\frac{\dot f_k(t_{k-1})}{j\omega}+\frac{\dot f_k^{(')}(t_{k-1})}{(j\omega)^2}+\right.$$

$$\left.+\frac{\dot f_k^{('')}(t_{k-1})}{(j\omega)^3}+\ldots\right]e^{-j\omega t_{k-1}},$$

откуда согласно (5.46)

$$S_k(\omega; t_{k-1}) = \sum_{i=0}^{\infty} \frac{f_{k-1}^{(i)}(t_{k-1}) - f_k^{(i)}(t_{k-1})}{(j\omega)^{i+1}} e^{-j\omega t_{k-1}}. \quad (5.47)$$

Полученная формула определяет мгновенный спектр комплексной функции в точке t_{k-1} локальными свойствами функции в этой точке. Если в рассматриваемой точке все производные функции (включая нулевую) непрерывны, то мгновенный спектр равен нулю. Если число членов ряда (5.47) конечно или если ряд суммируется, то в результате получается точное значение спектра. Общая спектральная функция процесса определяется суммой мгновенных спектров (5.47) для всех точек нарушения гладкости комплексной огибающей t_k (см. рис. 5.1).

Такое представление обладает рядом существенных достоинств. Во-первых, при аппроксимации процесса число слагаемых в выражении (5.47) невелико: квадратичная парабола ($i_{max}=2$) обычно обеспечивает требуемую точность аппроксимации. Во-вторых, благодаря использованию аналитических свойств спектральной функции обеспечивается возможность точного спектрального анализа при наличии разрывов функции и (или) ее производных. В-третьих, реализуется возможность адаптивного представления свойств функции: на участках быстрого изменения отсчеты функции располагаются близко, на плавных участках — далеко друг от друга. Наконец, общая спектральная функция строится последовательным суммированием мгновенных спектров (5.47), т. е. возможно вычисление текущего спектра функции. Для сравнения отметим, что расчет методом ДПФ для получения текущего спектра требует повторения вычислений по формуле (5.42) для последовательно возрастающих участков процесса. Кроме того, согласно выражению (5.47) значения спектральной функции могут рассчитываться по частоте с произвольным шагом, не связанным с числом использованных точек отсчета.

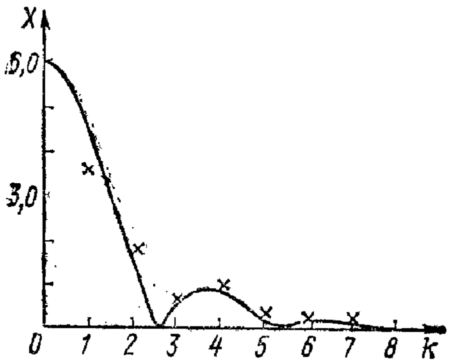


Рис. 5.2. Модуль спектральной функции трапецеидального видеоимпульса

Рассмотрим, например, изменение модуля спектральных функций симметричного трапецеидального видеоимпульса единичной амплитуды (рис. 5.2) при соотношении длительности фронта к длительности импульса, равном $t_ф/t_и=1/3$, рассчитанные с помощью ДПФ при $N=16$ (крестики) и с использованием выражения (5.47). Шаг расчета по частоте выбран достаточно малым, что позволяет считать полученные с помощью (5.47) результаты точными, поскольку линейная аппроксимация точно представляет трапецеидальный импульс. Из рис. 5.2 видно, что можно с помощью (5.47) определить такие важные характеристики сигнала, как нули спектральной функции. Вычисления с помощью ДПФ дают заметную погрешность, хотя число точек отсчета $N=16$ довольно велико. Все это позволяет считать, что использование полученных соотношений при расчете спектральных функций может быть весьма эффективным.

Глава шестая

СТАТИСТИЧЕСКИЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ВЕЛИЧИН

6.1. Статистическое оценивание законов распределения случайных величин

Рассмотрим методы построения эмпирических (статистических) законов распределения случайных величин, которые служат, во-первых, в качестве оценок истинных, но неизвестных законов распределения, во-вторых, для сжатия экспериментальных данных, поскольку по сгруппированным данным при большом объеме выборки существенно проще оценивать параметры распределения (см. § 6.2).

Дадим основные определения. Пусть получена выборка $\{x_i\}$ объема n дискретной или непрерывной случайной величины X. Последовательность значений выборки, записанная в возрастающем (не убывающем) порядке называется вариационным рядом.

Эмпирическим (статистическим) распределением выборки дискретной случайной величины называют перечень возможных значений случайной величины x_i из интервала значений, занимаемых вариационным рядом, и соответствующих им абсолютных частот m_i (сумма всех частот равна объему выборки) или относительных частот w_i (сумма всех относительных частот равна единице). Полигоном частот называют ломаную, отрезки которой соединяют точки $\{x_i, m_i\}$ или $\{x_i, w_i\}$ ($i=1, ..., M$).

Эмпирическим (статистическим) распределением выборки непрерывной случайной величины называют последовательность частичных интервалов $\{x_i, x_{i+1}\}$, на которые разбивается интервал, занимаемый вариационным рядом, и соответствующих им абсолютных m_i или относительных w_i частот. Гистограммой частот называют ступенчатую фигуру, состоящую из прямоугольников, основаниями которых служат частичные интервалы

$\{x_i, x_{i+1}\}$, имеющие длину $\{h_i\}$, а высота равна отношению $\{m_i/h_i\}$ (плотность частоты). Площадь i-го частичного прямоугольника равна величине m_i, площадь гистограммы равна объему выборки. Иногда рассматривают гистограмму относительных частот — ступенчатую фигуру, отличающуюся от предыдущей тем, что высота прямоугольников равна отношению $\{\omega_i/h_i\}$ (плотность относительной частоты). Площадь i-го частичного прямоугольника равна величине ω_i, а площадь всей гистограммы — единице.

Ввиду ограниченных возможностей Б3-21 приведем программу для определения абсолютных частот $\{m_i\}$ лишь для Б3-34. Интервал возможных значений непрерывной случайной величины $[x_{\min}, x_{\max}]$ разбивается на $k \leqslant 10$ равных интервалов. Программа 6.1/34 взята из работы [38]:

Программа 6.1/34

—	С/П	П0	ПВ	1	ВП	6	/—/	FBx	F→
—	÷	ПС	÷	1	—	ПД	Сx	КП↑	FLO
18	С/П	ИПС	÷	ИПД	—	П0	КИП↑	1	+
КП↑	БП	21	ИПВ	П0	КИП↑	С/П	FLO	35	

Инструкция к программе 6.1/34: В/О; набрать $x_{\min}$↑; $x_{\max}$ С/П; k С/П; поочередно набирать x_i С/П; после ввода всех данных БП 33; нажимая клавишу С/П, поочередно вывести m_j $(j=1, \dots, k)$, которые помещаются в РХ.

6.2. Точечные оценки параметров распределений случайных величин и процессов

Закон распределения (в любой его форме) является исчерпывающей характеристикой случайной величины. Однако задача его определения является достаточно сложной. Поэтому на практике часто определяют не законы распределения, а их числовые характеристики или параметры, основными из которых являются (при неявном допущении гауссовости) математическое ожидание, дисперсия и коэффициент корреляции. В связи с этим рассмотрим выражения для оценок числовых характеристик одномерных и двухмерных случайных (гауссовских) величин. Поскольку сечения случайных процессов могут рассматриваться как случайные величины, то здесь же приведем выражения для оценок математического ожидания, дисперсии и корреляционной функции процесса (табл. 6.1). В таблице обозначено: x_i и $x_i(t_j)$ — i-я реализация случайной величины X или случайного процесса $X(t)$ в момент времени t_j, n — число реализаций, N — число временных отсчетов случайного процесса.

Программы 6.2 предназначены для определения выборочного среднего $m \equiv \hat{X}$ и выборочной дисперсии $s^2 \equiv \hat{\sigma}^2_X$ и корня квадратного из выборочной дисперсии $s = \sqrt{s^2}$, который является состоятельной, хотя и смещенной оценкой среднего квадратического отклонения (см. ниже):

Программа 6.2/21

С/П	Р2	↑	F4	+	Р4	F5	↑	F2	Fx^2	+	Р5
F3	↑	1	—	Р3	РХ=0	Р0	F4	↑	F7	÷	Р4
F7	↑	1	—	Р3	F4	Fx^2	↑	F7	×	/—/	↑
F5	+	↑	F3	÷	Р5	F$\sqrt{}$	Р6	С/П			

Программа 6.2/34

П0	П1	0	П2	П3	С/П	↑	ИП2	+	П2
XY	Fx^2	ИП3	+	П3	FLO	05	ИП2	ИП1	÷
П4	Fx^2	ИП1	×	ИП3	XY	—	КИП1	XY	ИП1
÷	П5	F$\sqrt{}$	С/П						

Инструкция к программе 6.2/21: В/О; набрать 0 Р4 Р5 Р6; набрать n Р3; Р7: поочередно набирать x_i С/П; С/П; при этом текущее значение $n-i$ находится в РХ; в результате расчета m содержится в Р4, s^2 — в Р5, s — в Р6.

Инструкция к программе 6.2/34: В/О; набрать n С/П; поочередно набирать x_i С/П; после ввода последнего значения величина s содержится в РХ, $(n-1)$ — в Р1, Σx_i — в Р2, Σx_i^2 — в Р3, m — в Р4, s^2 — в Р5.

Иногда требуется определять числовые характеристики не по выборке полного объема, а по мере поступления новых данных. Программы 6.3 позволяют рассчитывать текущее выборочное среднее в соответствии с выражением

$$m_i = m_{i-1} + \frac{x_i - m_{i-1}}{i}.$$

Таблица 6.1

Оцениваемая числовая характеристика	Выражение для оценки
Математическое ожидание $\bar{X}$ случайной величины X и сечения случайного процесса $X(t)$	$\hat{X} \equiv m = \frac{1}{n}\sum_{i=1}^{n} x_i$; $\hat{X}(t_j) = \frac{1}{n}\sum_{i=1}^{n} x_i(t_j)$; для эргодических процессов $\hat{X} = \frac{1}{N}\sum_{j=1}^{N} x_i(t_j)$
Дисперсия случайной величины и сечения случайного процесса σ_X^2	$\hat{\sigma}_X^2 \equiv s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \hat{X}\right)^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right]$; $\hat{\sigma}_X^2(t_j) = \frac{1}{n-1}\sum_{i=1}^{n}\left[x_i(t_j) - \hat{X}(t_j)\right]^2$; для эргодических процессов $\hat{\sigma}_X^2 = \frac{1}{N-1}\sum_{j=1}^{N}\left[x_i(t_j) - \hat{X}\right]^2$
Корреляционный момент R_{XY}	$\hat{R}_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \hat{X}\right)\left(y_i - \hat{Y}\right) = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)\right]$

Продолжение табл. 6.1

Оцениваемая числовая характеристика	Выражение для оценки
Коэффициент корреляции ρ_{XY}	$\hat{\rho}_{XY} \equiv r_{XY} = \frac{\hat{K}_{XY}}{\sqrt{\hat{\sigma}_X^2 \hat{\sigma}_Y^2}}$
Взаимная ковариационная функция $K_{XY}(t_m, t_k)$ процессов $X(t)$ и $Y(t)$	$\hat{K}_{XY}(t_m, t_k) = \frac{1}{n-1}\sum_{i=1}^{n} x_i(t_m)\, y_i(t_k)$
Ковариационная функция $K_X(t_m, t_k)$ процесса $X(t)$	$\hat{K}_X(t_m, t_k) = \frac{1}{n-1}\sum_{i=1}^{n} x_i(t_m)\, x_i(t_k)$
Корреляционная функция $R_X(t_m, t_k)$ процесса $X(t)$	$\hat{R}_X(t_m, t_k) = \frac{1}{n-1}\left\{\sum_{i=1}^{n}\left[x_i(t_m) - \hat{X}(t_m)\right]\left[x_i(t_k) - \hat{X}(x_k)\right]\right\} = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i(t_m)\, x_i(t_k) - \frac{1}{n}\left(\sum_{i=1}^{n} x_i(t_m)\right)\left(\sum_{i=1}^{n} x_i(t_k)\right)\right]$
Нормированная корреляционная функция $\rho_X(t_m, t_k)$ процесса $X(t)$	$\hat{\rho}_X(t_m, t_k) \equiv r_X(t_m, t_k) = \frac{\hat{R}_X(t_m, t_k)}{\sqrt{\hat{\sigma}_X^2(t_m)\, \hat{\sigma}_X^2(t_k)}}$

Оцениваемая числовая характеристика	Выражение для оценки
Параметры уравнения регрессии случайной величины Y по случайной величине X $Y=bX+a$	$\hat{b}=\left[n\sum\limits_{i=1}^{n}x_iy_i-\left(\sum\limits_{i=1}^{n}x_i\right)\left(\sum\limits_{i=1}^{n}y_i\right)\right]:$ $:\left[n\sum\limits_{i=1}^{n}x_i^2-\left(\sum\limits_{i=1}^{n}x_i\right)^2\right];$ $\hat{a}=\hat{Y}-\hat{b}\hat{X}$
Параметры уравнения регрессии случайной величины X по случайной величине Y $X=b'Y+a'$	$\hat{b}'=\left[n\sum\limits_{i=1}^{n}x_iy_i-\left(\sum\limits_{i=1}^{n}x_i\right)\left(\sum\limits_{i=1}^{n}y_i\right)\right]:$ $:\left[n\sum\limits_{i=1}^{n}y_i^2-\left(\sum\limits_{i=1}^{n}y_i\right)^2\right];$ $\hat{a}'=\hat{X}-\hat{b}'\hat{Y}$

Программа 6.3/21

Р6	1	Р7	0	Р8	F6	↑	F8	—	↑	F7	÷
↑	F8	+	Р8	F7	↑	1	+	Р7	F8	С/П	БП
Р1											

Программа 6.3/34

П0	0	П1	1	П4	ИП0	ИП1	—	ИП4	÷
ИП1	+	П1	С/П	П0	КИП4	БП	05		

Следует отметить, что текущие оценки более устойчивы к ошибкам, вызванным переполнением разрядной сетки.

Инструкция к программам 6.3: В/О; поочередно набирать x_i С/П; m содержится в РХ (в калькуляторе Б3-21 также в Р8), в Р4 калькулятора Б3-34 содержится значение i, в Р4 Б3-21 — значение $i+1$.

Вернемся к вопросу об оценке среднего квадратического значения. Известно, что несмещенная и состоятельная оценка имеет вид

$$s'=K_n s=\sqrt{\frac{n-1}{n}}\,\frac{\Gamma\left(\frac{n-1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}s,$$

где Γ — гамма-функция.

При $n\geqslant 10$ можно использовать приближенное выражение

$$s'\approx\left(1+\frac{1}{4n}+\frac{9}{32n^2}\right)s. \qquad (6.1)$$

Для $4\leqslant n\leqslant 9$ целесообразно пользоваться следующими данными:

n	4	5	6	7	8	9
K_n	1.085	1.064	1.051	1.042	1.036	1.032

Однако выражение (6.1) часто можно применять и при $4\leqslant n\leqslant 10$. При этом погрешность не будет превышать 0,5 %. Этот алгоритм реализован в программах 6.4:

Программа 6.4/21

Р2	С/П	Р3	4	×	F1/x	Р4	F3	Fx^2	3	2	×
↑	9	÷	F1/x	↑	F4	+	1	+	Р4	↑	F2
F$\sqrt{}$	×	Р5	С/П								

Программа 6.4/34

F$\sqrt{}$	П0	С/П	П1	4	+	F1/x	1	+	9
ИП1	Fx^2	3	2	×	÷	+	П2	ИП0	×
С/П									

Инструкция к программам 6.4: В/О; набрать s^2 С/П; n С/П; s' содержится в РХ (в Б3-21 также в Р5), K_n — в Р4 (в Б3-34 — в Р2).

В некоторых случаях выборка $\{x_i\}$ $(i=1,\ldots,n)$ объема n задана в виде сгруппированных данных, т. е. в виде распределения средних точек $\{\overline{x_i}\}$ $(i=1,\ldots,M)$ M интервалов, на которые разбита область возможных значений случайной величины X, и соответствующих абсолютных частот $\{m_i\}$ $(i=1,\ldots,M)$.

В этом случае выборочное среднее и выборочную дисперсию находят с помощью выражений

$$m \equiv \hat{\bar{X}} = \frac{1}{n}\sum_{i=1}^{M} m_i \bar{x}_i;$$

$$s^2 \equiv \hat{\sigma}_X^2 = \frac{1}{n-1}\left[\sum_{i=1}^{M} m_i \bar{x}_i^2 - \frac{1}{n}\left(\sum_{i=1}^{M} m_i x_i\right)^2\right].$$

Если число интервалов M мало и они имеют одинаковую длину $h = \bar{x}_{i+1} - \bar{x}_i$ $(i=1, \ldots, M-1)$, для уменьшения ошибки определения выборочной дисперсии, вызванной группировкой данных, используют так называемую поправку Шеппарда [21]:

$$s_1^2 = s^2 - h^2/12.$$

Этот алгоритм реализован в программах 6.5:

Программа 6.5/21

Fx^2	1	2	÷	Р2	С/П	Р3	С/П	↑	Р4	F3	×
↑	F6	+	Р6	F3	×	↑	F7	+	Р7	F4	↑
F8	+	Р8	С/П	БП	Р1	F6	↑	F8	÷	Р3	F6
Fx^2	↑	F8	÷	/—/	↑	F7	+	Р4	F8	1	—
$F1/x$	↑	F4	×	Р4	↑	F2	—	Р5	F3	С/П	

Программа 6.5/34

П0	0	П1	П2	П3	С/П	П4	ИП1	+	П1
С/П	П5	ИП4	×	ИП3	+	П3	ИП4	ИП5	Fx^2
×	ИП2	+	П2	БП	05	ИП2	ИП3	Fx^2	ИП1
÷	—	ИП1	1	—	÷	П6	ИП0	Fx^2	1
	÷	—	П7	ИП3	ИП1	÷	С/П		

Инструкция к программе 6.5/21: В/О; набрать 0 Р6 Р7 Р8; h С/П; попарно вводить x_i С/П; m_i С/П $(i=1, \ldots)$; после ввода всех данных набрать БП Р5 С/П; m содержится в РХ и Р3, s^2 — в Р4, s_1^2 — в Р5.

Инструкция к программе 6.5/34: В/О; набрать h С/П; попарно вводить m_i С/П; x_i С/П; после ввода всех данных набрать БП 26 С/П; m содержится в РХ, s^2 — в Р6, s_1^2 — в Р7.

Для расчета значений оценки корреляционной функции случайного процесса $\hat{R}_X(t_m, t_k)$ для моментов времени t_m и t_k используются программы 6.6:

Программа 6.6/21

Р5	С/П	↑	Р→	+	Р→	XY	Р6	Fx^2	+	Р→	↑
F5	+	Р→	↑	F5	Fx^2	+	Р→	Р7	F5	↑	F6
×	↑	F7	+	Р→	↑	1	+	Р2	Р→	С/П	БП
Р0	F2	↑	1	—	Р3	Р→	↑	Р→	Р→	Р4	×
↑	F2	÷	/—/	↑	F4	Р→	Р→	+	↑	F3	÷

Программа 6.6/34

0	П0	П1	П3	П4	П5	П6	С/П	ПД	ИП1
+	П1	ИПД	Fx^2	ИП4	+	П4	С/П	ПС	ИП0
+	П0	ИПС	Fx^2	ИП3	+	П3	ИПД	ИПС	×
ИП6	+	П6	КИП5	БП	07	ИП5	П2	КИП2	ИП1
ИП5	÷	П7	ИП0	ИП5	÷	П8	ИП3	ИП1	Fx^2
ИП5	÷	—	ИП2	÷	П9	ИП4	ИП0	Fx^2	ИП5
÷	—	ИП2	÷	ПА	ИП6	ИП0	ИП1	×	ИП5
÷	—	ИП2	÷	ПД	ИП7	ИП8	×	+	ПВ
ИПД	ИП9	ИПА	×	$F\sqrt{}$	÷	ПС	ИПД	С/П	

Программы позволяют также определять дополнительные параметры:

$$A = \sum_i x_i(t_m); \quad B = \sum_i x_i^2(t_m);$$

$$C = \sum_i x_i(t_k); \quad D = \sum_i x_i^2(t_k).$$

Эти величины, как видно из табл. 6.1, позволяют при необходимости определить оценки математических ожиданий и дисперсий сечений случайного процесса $X(t_m)$ и $X(t_k)$.

Инструкция к программе 6.6/21: В/О; очистить стек, для чего набрать 0 Р→; 0 Р→; 0 Р→; 0 Р→; 0 Р→; 0 Р→; попарно вводить значения $x_i(t_m)$ С/П; $x_i(t_k)$ С/П $(i=1, \ldots, n)$; после ввода всех данных набрать БПF6 С/П; значение $\hat{R}_X(t_m, t_k)$ содержится в РХ и Р5; для вывода дополнительных данных последовательно набирать Р→, при этом из стека в РХ выводятся значения n, $n-1$, C, D, A, B.

Инструкция к программе 6.6/34: В/О; С/П; попарно набирать $x_i(t_m)$ С/П; $x_i(t_k)$ С/П; после ввода всех данных значение A содержится в Р1, B — в Р3, C — в Р0, D — в Р4, n — в Р5; для окончания расчетов набрать БП 36 С/П; $\hat{R}_X$ — в РХ, $\hat{\rho}_X$ — в РС, $\hat{K}_X$ — в РВ, $\hat{\sigma}_X^2(t_m)$ — в Р9, $\hat{\sigma}_X^2(t_k)$ — в РА, $m(t_m)$ — в Р7, $m(t_k)$ — в Р8, $(n-1)$ — в Р2.

Из табл. 6.1 также видно, что эти же программы можно использовать для определения оценки корреляционного момента случайных величин X и Y. Для этого достаточно принять $X(t_m)=X$, $X(t_k)=Y$.

Контрольный пример. Исходные данные: $\{x_i(t_m),$ $x_i(t_k)\}=1{,}1;$ 2,2, 3,3, 4,4, 5,5. Результаты: $\hat{R}_X(t_m,\ t_k)=$ $=2{,}5$; $A=15$; $B=55$; $C=15$; $D=55$.

6.3. Интервальные оценки параметров распределений

Интервальные оценки для вероятности события. Предварительно напомним основные понятия теории интервального оценивания.

Степень близости статистической оценки $\hat{\theta}$ к соответствующему параметру распределения θ удобно характеризовать с помощью доверительного интервала $J=[\theta_н,\ \theta_в]$, удовлетворяющего равенству

$$P[\theta \in J]=\varepsilon.$$

Величина ε — доверительная вероятность, $\theta_н$ и $\theta_в$ — нижняя и верхняя доверительные границы, которые непосредственно связаны с точечной оценкой параметра $\hat{\theta}$.

Доверительный интервал $J_i[p_{нi},\ p_{вi}]$ $(i=1,\ 2,\ 3)$ для неизвестной вероятности p появления события A в каждом испытании строится по его абсолютной частоте m наблюдения случайного события в n независимых испытаниях.

Пусть Z — случайное число появлений события A в n испытаниях. Известно [7], что эта случайная величина имеет биномиальное распределение с параметрами n и p: $B(n,\ p)$.

В зависимости от постановки задачи устанавливаются следующие доверительные интервалы [19]:

а) двухсторонний доверительный интервал $J_1=$ $=[p_{н1},\ p_{в1}]$ с доверительными границами

$$p_{н1}=1-\frac{(n-k+1)\,F_{2(n-k+1),\ 2k;\ (1+\varepsilon)/2}}{k+(n-k+1)\,F_{2(n-k+1),\ 2k;\ (1+\varepsilon)/2}};$$

$$p_{в1}=\frac{(k+1)\,F_{2(k+1),\ 2(n-k);\ (1+\varepsilon)/2}}{n-k+(k+1)\,F_{2(k+1),\ 2(n-k);\ (1+\varepsilon)/2}};$$

б) односторонний доверительный интервал $J_2=$ $=[p_{н2},\ 1]$ с нижней границей

$$p_{н2}=1-\frac{(n-k+1)\,F_{2(n-k+1),\ 2k;\ \varepsilon}}{k+(n-k+1)\,F_{2(n-k+1),\ 2k;\ \varepsilon}};$$

в) односторонний доверительный интервал $J_3=$ $=[0,\ p_{в3}]$ с верхней доверительной границей

$$p_{в3}=\frac{(k+1)\,F_{2(k+1),\ 2(n-k);\ \varepsilon}}{n-k+(k+1)F_{2(k+1),\ 2(n-k);\ \varepsilon}}.$$

Здесь $F_{\nu_1\nu_2 q}$ — квантиль q-го порядка F-распределения с $(\nu_1,\ \nu_2)$ степенями свободы (см. табл. П.8).

Для расчета доверительных интервалов предназначены программы 6.7.

Программа 6.7/21

P7	С/П	P3	↑	F7	—	P2	С/П	P4	ПП	/—/	1
—	/—/	P6	С/П	P4	ПП	/—/	1	—	/—/	P8	С/П
P4	F7	P2	ПП	/—/	P7	С/П	P4	ПП	/—/	С/П	F2
1	+	↑	F4	×	P5	↑	F2	—	↑	F3	+
↑	F5	XY	÷	В/О							

Программа 6.7/34

П0	С/П	П1	С/П	П2	С/П	П3	С/П	П4	С/П
П5	ИП2	ПП	27	П6	ИП3	ПП	41	П7	ИП4
ПП	27	П8	ИП5	ПП	41	С/П	ИП1	ИП0	—
1	+	×	↑	ИП0	+	÷	1	XY	—
В/О	ИП0	1	+	×	↑	ИП1	ИП0	—	+
÷	В/О								

Инструкция к программе 6.7/21: В/О; набрать k С/П; n С/П; $F_{2(n-k+1),\ 2k;(1+\varepsilon)/2}$ С/П; $F_{2(n-k+1),\ 2k;\varepsilon}$ С/П; $F_{2(k+1),2(n-k);(1+\varepsilon)/2}$ С/П; $F_{2(k+1),2(n-k);\varepsilon}$ С/П; $p_{н1}$ содержится в Р6, $p_{в1}$ — в Р7, $p_{н2}$ — в Р8, $p_{в3}$ — в РХ.

Инструкция к программе 6.7/34: В/О; набрать k С/П; n С/П; $F_{2(n-k+1),2k;\ (1+\varepsilon)/2}$ С/П; $F_{2(k+1),\ 2(n-k);\ (1+\varepsilon)/2}$ С/П; $F_{2(n-k+1),2k;\ \varepsilon}$ С/П; $F_{2(k+1),\ 2(n-k);\varepsilon}$ С/П; расположение результатов в регистрах аналогично предыдущему.

Контрольный пример. Исходные данные: $k=3$; $n=$ $=10$; $F_{16,6;\ 0{,}975}=5{,}24$; $F_{16,6;\ 0{,}95}=3{,}92$; $F_{8,14;\ 0{,}975}=3{,}29$; $F_{8,14;\ 0{,}95}=2{,}70$. Результаты: $p_{н1}=6{,}67854\cdot10^{-2}$; $p_{в1}=$ $=6{,}527777\cdot10^{-1}$; $p_{н2}=8{,}73109\cdot10^{-2}$; $p_{в3}=6{,}067415\cdot10^{-1}$.

При большом числе испытаний $(n\geqslant30)$ удобнее рассчитывать асимптотический двухсторонний интервал $J=[p_н,\ p_в]$, который имеет границы [19]:

$$p_{н} = \frac{1}{n + \lambda^2_{(1+\varepsilon)/2}} \left(k + \frac{\lambda^2_{(1+\varepsilon)/2}}{2} - \lambda_{(1+\varepsilon)/2} \times \right.$$

$$\left. \times \sqrt{\frac{k(n-k)}{n} + \frac{\lambda^2_{(1+\varepsilon)/2}}{4}} \right);$$

$$p_{в} = \frac{1}{n + \lambda^2_{(1+\varepsilon)/2}} \left(k + \frac{\lambda^2_{(1+\varepsilon)/2}}{2} + \right.$$

$$\left. + \lambda_{(1+\varepsilon)/2} \sqrt{\frac{k(n-k)}{n} + \frac{\lambda^2_{(1+\varepsilon)/2}}{4}} \right),$$

где λ_q — квантиль порядка q стандартного гауссовского распределения (см. табл. П.4).

Этот доверительный интервал определяется с помощью программ 6.8:

Программа 6.8/21

Р2	Fx^2	Р3	С/П	Р4	+	F1/x	Р5	С/П	Р6	↑	F4
—	/—/	×	↑	F4	÷	Р7	F3	4	÷	↑	F7
+	F√	↑	F2	×	Р8	F3	2	÷	↑	F6	+
Р7	↑	F8	—	↑	F5	×	Р3	F7	↑	F8	+
↑	F5	×	Р4	С/П							

Программа 6.8/34

П0	Fx^2	П1	С/П	П2	С/П	П3	—	ИП3	×
ИП2	÷	ИП1	4	÷	+	F√	ИП0	×	↑
ИП1	2	÷	ИП3	+	П4	+	XY	ИП4	XY
—	ИП1	ИП2	+	П4	÷	П5	XY	ИП4	÷
С/П									

Инструкция к программам 6.8: В/О; набрать $\lambda_{(1+\varepsilon)/2}$ С/П; n С/П; k С/П; в Б3-21 $p_в$ содержится в РХ и Р4, $p_н$ — в Р3, в Б3-34 $p_в$ содержится в РХ, $p_н$ — в Р5.

Контрольный пример. Исходные данные: $k=4$; $n=30$; $\lambda_q=1{,}96$ $(\varepsilon=0{,}95)$. Результаты: $p_н=5{,}309569 \times 10^{-2}$; $p_в=2{,}968168 \cdot 10^{-1}$.

Интервальные оценки для среднего значения гауссовской случайной величины. Доверительные интервалы строятся с доверительной вероятностью ε по результатам выборки $\{x_i\}$ объема n, по которой вычислено выборочное среднее m.

В зависимости от постановки задачи устанавливаются двухсторонний и односторонние интервалы.

Рассмотрим случай, когда известна дисперсия случайной величины σ^2:

а) двухсторонний доверительный интервал $J_1=[\bar{x}_{н1}, \bar{x}_{в1}]$ имеет границы

$$\bar{x}_{н1} = m - \lambda_{(1+\varepsilon)/2}\, \sigma/\sqrt{n};$$

$$\bar{x}_{н1} = m + \lambda_{(1+\varepsilon)/2}\, \sigma/\sqrt{n};$$

б) односторонний доверительный интервал $J_2=(-\infty, \bar{x}_{в2}]$ имеет верхнюю доверительную границу

$$\bar{x}_{в2} = m + \lambda_{\varepsilon}\, \sigma/\sqrt{n};$$

в) односторонний доверительный интервал $J_3=[\bar{x}_{н3}, \infty)$ имеет нижнюю доверительную границу

$$\bar{x}_{н3} = m - \lambda_{\varepsilon}\, \sigma/\sqrt{n}.$$

Здесь λ_q — квантиль порядка q стандартного гауссовского распределения (см. табл. П.4).

Программы 6.9 позволяют определить доверительные границы непосредственно по выборке $\{x_i\}$, $i=1, ..., n$:

Программа 6.9/21

↑	F2	+	Р2	F3	↑	1	+	Р3	С/П	БП	Р0
F2	↑	F3	÷	Р2	С/П	Р4	С/П	Р5	С/П	Р7	РНОП
F4	↑	F3	F√	÷	Р3	↑	F5	×	↑	F2	+
Р6	XY	/—/	↑	F2	+	Р5	F3	↑	F7	×	↑
F2	+	Р8	XY	/—/	↑	F2	+	Р7	С/П		

Программа 6.9/34

П0	С/П	П1	С/П	П2	0	П3	П4	С/П	ИП3
+	П3	КИП4	БП	08	ИП3	ИП4	÷	П5	ИП0
ИП4	F√	÷	П6	ИП5	ИП1	ИП6	×	—	П7
ИП5	ИП1	ИП6	×	+	П8	ИП5	ИП2	ИП6	×
—	П9	ИП5	ИП2	ИП6	×	+	С/П		

Инструкция к программе 6.9/21: В/О; набрать 0 Р2 Р3; поочередно набирать x_i С/П; при этом в РХ и Р3 содержится номер введенного отсчета i, в Р2 — текущая сумма $\sum_{j=1}^{i} x_j$; после ввода последнего значения набрать БП Р2 С/П; в Р2 содержится m, в Р3 — n; набрать σ С/П; $\lambda_{(1+\varepsilon)/2}$ С/П; λ_ε С/П; $\bar{x}_{н1}$ содержится в Р5, $\bar{x}_{в1}$ — в Р6, $\bar{x}_{н2}$ — в Р7, $\bar{x}_{в3}$ — в Р8.

Инструкция к программе 6.9/34: В/О; набрать σ С/П; $\lambda_{(1+\varepsilon)/2}$ С/П; λ_ε С/П; поочередно набирать x_i С/П; после ввода последнего значения набрать БП 15 С/П; m содержится в Р5, n — в Р4, $\bar{x}_{в3}$ — в РХ, $\bar{x}_{н2}$ — в Р9, $\bar{x}_{в1}$ — в Р8, $\bar{x}_{н1}$ — в Р7.

Программы 6.10 применяются в случае, когда выборочное среднее вычислено предварительно:

Программа 6.10/21

Р2	С/П	Р3	С/П	Р4	С/П	Р5	С/П	Р7	РНОП	F4	↑
F3	F√	÷	Р3	↑	F5	×	↑	F2	+	Р6	XY
/—/	↑	F2	+	Р5	F3	↑	F7	×	↑	F2	+
Р8	XY	/—/	↑	F2	+	Р7	С/П				

Программа 6.10/34

П0	С/П	↑	С/П	F√	÷	П1	С/П	П2	С/П
ИП1	×	П3	ИП0	+	П4	ИП0	ИП3	—	П5
ИП2	ИП1	×	П3	ИП0	+	П6	ИП0	ИП3	—
С/П									

Инструкция к программам 6.10: В/О; набрать m С/П; σ С/П; n С/П; $\lambda_{(1+\varepsilon)/2}$ С/П; λ_ε С/П; $\bar{x}_{н1}$ содержится в Р5 (РХ), $\bar{x}_{в1}$ — в Р6, $\bar{x}_{н2}$ — в Р7 (Р5), $\bar{x}_{в3}$ — в Р8 (Р4) (в скобках указаны регистры для микрокалькулятора Б3-34).

В тех случаях, когда дисперсия случайной величины неизвестна [6, 19], используются следующие доверительные интервалы:

а) двухсторонний доверительный интервал $J_1=[\bar{x}_{н1}, \bar{x}_{в1}]$ с границами

$$\bar{x}_{н1} = m - s/\sqrt{n}\, t_{n-1;\,(1+\varepsilon)/2};$$

$$\bar{x}_{в1} = m + s/\sqrt{n}\, t_{n-1;\,(1+\varepsilon)/2};$$

б) односторонний доверительный интервал $J_2 = (-\infty, \bar{x}_{в2}]$ с верхней доверительной границей

$$\bar{x}_{в2} = m + st_{n-1;\varepsilon}/\sqrt{n};$$

в) односторонний доверительный интервал $J_3 = [\bar{x}_{н3}, \infty)$ с нижней доверительной границей

$$\bar{x}_{н3} = m - st_{n-1;\varepsilon}/\sqrt{n}.$$

Здесь s^2 — выборочная дисперсия; $t_{\nu q}$ — квантиль q-го порядка t-распределения с ν степенями свободы (см. табл. П.5).

Используется два варианта программ. Если среднее выборочное значение и выборочная дисперсия вычислены заранее, то для определения доверительных интервалов используются программы 6.10. При этом в них вместо величин σ, $\lambda_{(1+\varepsilon)/2}$ и λ_ε используются величины s, $t_{n-1;\,(1+\varepsilon)/2}$ и $t_{n-1;\varepsilon}$ соответственно. В тех случаях, когда доверительные интервалы необходимо определить непосредственно по выборке $\{x_i\}$, используются программы 6.11:

Программа 6.11/21

↑	F2	+	Р2	F3	XY	Fx^2	+	Р3	F4	1	+
Р4	XY	Р5	F4	С/П	БП	Р0	F2	↑	F4	÷	Р2
Fx^2	↑	F4	×	/—/	↑	F3	+	↑	F5	÷	Р3
F5	С/П	F3	↑	F4	÷	F√	↑	F6	×	↑	F2
+	Р8	XY	/—/	↑	F2	+	Р7	С/П	БП	РВП	

Программа 6.11/34

0	П0	П1	П4	С/П	↑	ИП0	+	П0	XY
Fx^2	ИП1	+	П1	КИП4	ИП4	БП	04	П6	С/П
П7	ИП0	ИП4	÷	П2	Fx^2	ИП4	×	ИП1	XY
—	ИП4	1	—	÷	П3	ИП4	÷	F√	П5
ИП6	×	ИП2	+	П8	ИП2	ИП6	ИП5	×	—
П9	ИП2	ИП7	ИП5	×	+	ПА	ИП2	ИП7	ИП5
×	—	С/П							

Инструкция к программе 6.11/21: В/О; набрать 0 Р2 Р3 Р4; поочередно набирать x_i С/П; при этом в Р2 содержится $\sum_{j=1}^{i} x_j$, в Р3 — $\sum_{j=1}^{i} x_j^2$, в РХ и Р4 — i, в Р5 — $(i-1)$; после ввода последнего значения набрать БП F3 С/П, при этом в РХ содержится $(n-1)$, в Р2 — m, в Р3 — s^2; набрать $t_{n-1;(1+\varepsilon)/2}$; СП $(t_{n-1;\varepsilon})$ Р6 С/П; $\bar{x}_{н1}$ содержится в Р7, $\bar{x}_{в1}$ — в Р8 ($\bar{x}_{н2}$ — в Р7, $\bar{x}_{в3}$ — в Р8).

Инструкция к программе 6.11/34: В/О; С/П; поочередно набирать x_i С/П; после ввода последнего значения набрать БП 18; $t_{n-1;\,(1+\varepsilon)/2}$ С/П; $t_{n-1;\,\varepsilon}$ С/П; n содержится в Р4, m — в Р2, s^2 — в Р3, $\bar{x}_{н1}$ — в Р9, $\bar{x}_{в1}$ — в Р8, $\bar{x}_{н2}$ — в РХ, $\bar{x}_{в3}$ — в РА.

Интервальные оценки для дисперсии гауссовской случайной величины. Доверительные интервалы строятся с доверительной вероятностью ε по результатам выборки $\{x_i\}$ объема n, по которой вычисляется выборочная дисперсия s^2.

В зависимости от постановки задачи устанавливаются следующие доверительные интервалы [20]:

а) двухсторонний доверительный интервал $J_1 = [\sigma^2_{н1}, \sigma^2_{в1}]$ с границами

$$\sigma^2_{н1} = (n-1)\, s^2/\chi_{n-1;\ (1+\varepsilon)/2};$$

$$\sigma^2_{в1} = (n-1)\, s^2/\chi_{n-1;\ (1-\varepsilon)/2};$$

б) односторонний доверительный интервал $J_2 = [0, \sigma^2_{в2}]$ с верхней доверительной границей

$$\sigma^2_{в2} = (n-1)\, s^2/\chi^2_{n-1;\ 1-\varepsilon};$$

в) односторонний доверительный интервал $J_3 = [\sigma^2_{н3}, \infty]$ с нижней доверительной границей

$$\sigma^2_{н3} = (n-1)\, s^2/\chi^2_{n-1;\ \varepsilon}.$$

Здесь $\chi^2_{\nu;q}$ — квантиль q-го порядка χ^2-распределения с ν степенями свободы (см. табл. П.6).

Приведем два варианта программы расчета доверительных интервалов. Программы 6.12 используются в том случае, когда выборочная дисперсия s^2 вычислена предварительно:

Программа 6.12/21

Р2	С/П	$\uparrow$	1	—	Р3	С/П	Р4	С/П	Р5	С/П	Р6
С/П	Р7	РНОП	F2	$\uparrow$	F3	$\times$	$\uparrow$	F4	$\div$	Р4	F5
$\div$	Р5	F6	$\div$	Р6	F7	$\div$	Р7	С/П			

Программа 6.12/34

П0	С/П	П1	С/П	П2	С/П	П3	С/П	$\uparrow$	1
—	С/П	$\times$	П4	ИП0	$\div$	П5	ИП4	ИП1	$\div$
П6	ИП4	ИП2	$\div$	П7	ИП4	ИП3	$\div$	С/П	

Инструкция к программе 6.12/21: В/О; набрать s^2 С/П; n С/П; $\chi^2_{n-1;\ (1+\varepsilon)/2}$ С/П; $\chi^2_{n-1;\ (1-\varepsilon)/2}$С/П; $\chi^2_{n-1;\ 1-\varepsilon}$С/П; $\chi^2_{n-1;\ \varepsilon}$ С/П; $\sigma^2_{н1}$ содержится в Р4, $\sigma^2_{в1}$ — в Р5; $\sigma^2_{в2}$ — в Р6, $\sigma^2_{н3}$— в Р7.

Инструкция к программе 6.12/34: В/О; набрать $\chi^2_{n-1;\ (1+\varepsilon)/2}$ С/П; $\chi^2_{n-1;(1-\varepsilon)/2}$ С/П; $\chi^2_{n-1;\ 1-\varepsilon}$ С/П; $\chi^2_{n-1;\ \varepsilon}$С/П;$n$ С/П; s^2 С/П; $\sigma^2_{н1}$ содержится в Р5, $\sigma^2_{в1}$ — в Р6, $\sigma^2_{в2}$ — в Р7, $\sigma^2_{н3}$— в РХ.

Для вычисления доверительных интервалов и оценки дисперсии непосредственно по выборке $\{x_i\}$ служат программы 6.13:

Программа 6.13/21

$\uparrow$	F3	$+$	Р3	F4	XY	Fx^2	$+$	Р4	F5	Р2	1
$+$	Р5	С/П	БП	Р0	F3	Fx^2	$\uparrow$	F5	$\div$	$\uparrow$	F4
XY	—	$\uparrow$	F2	$\div$	Р3	С/П	Р4	С/П	Р5	С/П	Р6
С/П	Р7	РНОП	F3	$\uparrow$	F2	$\times$	$\uparrow$	F4	$\div$	Р4	F5
$\div$	Р5	F6	$\div$	Р6	F7	$\div$	Р7	С/П			

Программа 6.13/34

П2	0	П0	П1	П4	ИП2	$\uparrow$	Fx^2	ИП1	$+$
П1	XY	ИП0	$+$	П0	КИП4	С/П	БП	06	ИП1
ИП0	Fx^2	ИП4	$\div$	—	П0	ИП4	1	—	$\div$
П3	ИП0	С/П	$\div$	П8	ИП0	С/П	$\div$	П5	ИП0
С/П	$\div$	П6	ИП0	С/П	$\div$	П7	С/П		

Инструкция к программам 6.13: В/О; поочередно набирать x_i С/П; после ввода последнего значения набрать БП$\times$С/П (БП 19 С/П); в Р3 содержится значение s^2; набрать $\chi^2_{n-1;\ (1+\varepsilon)/2}$ С/П; $\chi^2_{n-1;\ (1-\varepsilon)/2}$ С/П; $\chi^2_{n-1;\ 1-\varepsilon}$ С/П; $\chi^2_{n-1;\ \varepsilon}$ С/П; $\sigma^2_{н1}$ содержится в Р4 (Р8), $\sigma^2_{в1}$ — в Р5, $\sigma^2_{в2}$ — в Р6, $\sigma^2_{н3}$ — в Р7 (в скобках указаны команды для Б3-34). В Б3-21 перед выполнением программы очистить регистры Р2 — Р5: набрать 0 Р2 Р3 Р4 Р5.

Интервальная оценка коэффициента корреляции двухмерной гауссовской случайной величины. Пусть двухмерная случайная величина (X, Y) имеет гауссовское распределение. Плотность распределения выборочного коэффициента корреляции r_{XY}, вычисляемого по результатам выборки $\{x_i, y_i\}$ объема n имеет весьма сложный вид. Поэтому для определения доверительных интервалов используются различные способы аппроксимации этого распределения или преобразования, позволяющие получить случайную величину с приблизительно гауссовским распределением. В частности, широко используется z-преобразование Фишера

$$z = \frac{1}{2}\ln\frac{1+r_{XY}}{1-r_{XY}}; \quad r_{XY} = \operatorname{th} z,$$

которое преобразует случайную величину r_{XY} к случайной величине z, имеющей приближенно распределение вида $N\left(\frac{1}{2}\ln\frac{1+\rho}{1-\rho} + \frac{\rho}{2(n-1)}, \frac{1}{n-3}\right)$ [5, 6, 19]. Здесь ρ —

истинное значение коэффициента корреляции, th — гиперболический тангенс:

$$\operatorname{th} z = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

Это преобразование используется даже в случае сравнительно небольших n ($n \geqslant 10$).

Двухсторонний доверительный интервал $J = [r_н, r_в]$ имеет границы [20]:

$$r_н = \operatorname{th}\left(z - \lambda_{(1+\varepsilon)/2} \frac{1}{\sqrt{n-3}}\right);$$

$$r_в = \operatorname{th}\left(z + \lambda_{(1+\varepsilon)/2} \frac{1}{\sqrt{n-3}}\right),$$

где λ_q — квантиль q-го порядка гауссовского распределения (см. табл. П.4).

Для расчета доверительного интервала коэффициента корреляции используются программы 6.14:

Программа 6.14/21

Р2	С/П	↑	3	—	F√	F1/x	Р3	С/П	↑	F3	×
Р3	F2	1	+	Р4	2	—	/—/	F1/x	↑	F4	×
Рln	2	÷	↑	F3	+	Р8	F3	—	ПП	ВП	Р7
F8	ПП	ВП	Р8	С/П	Р6	/—/	Рe^x	Р5	F6	Рe^x	↑
F5	+	Р6	F5	—	↑	F6	÷	В/О			

Программа 6.14/34

П0	СП	П1	С/П	П2	1	+	1	ИП2	—
÷	Fln	2	÷	П3	ИП1	ИП0	3	—	F√
÷	П4	—	ПП	32	П7	ИП3	ИП4	+	ПП
32	С/П	Fe^x	П5	↑	F1/x	П6	—	ИП5	ИП6
+	÷	В/О							

Инструкция к программе 6.14/21: В/О; набрать r_{XY} С/П; n С/П; $\lambda_{(1+\varepsilon)/2}$ С/П; значение $r_н$ содержится в Р7, $r_в$ — в РХ и Р8.

Инструкция к программе 6.14/34: В/О; набрать n С/П; $\lambda_{(1+\varepsilon)/2}$ С/П; r_{XY} С/П; значение $r_н$ содержится в РХ, $r_в$ — в Р7.

6.4. Статистическая проверка гипотез

Проверка гипотезы о распределении случайной величины по гауссовскому закону (критерий Пирсона). Нулевая гипотеза H_0 состоит в том, что случайная величина X распределена по гауссовскому закону.

В зависимости от формы представления выборки, по которой необходимо принять решение, и типа случайной величины (дискретная или непрерывная) возможны две модификации метода проверки гипотезы.

1. Пусть эмпирическое распределение задано в виде последовательности равноотносящих значений и соответствующих им абсолютных частот: (x_1, m_1; x_2, m_2; ... ...; x_M, m_M), где M — число различных возможных значений случайной величины.

Для проверки нулевой гипотезы при заданном уровне значимости α используется статистика

$$t = \sum_{i=1}^{M} (m_i - m_i')^2 / m_i,$$

где m_i' — теоретическая относительная частота значения x_i.

Величина m_i' вычисляется после определения по полученной выборке (x_1, m_1; ...; x_M, m_M) выборочного среднего значения m и выборочной дисперсии s^2 (например, с использованием программ 6.2) с помощью выражения

$$m_i' = \frac{nh}{\sqrt{s^2}} \varphi(u_i),$$

где $n = \Sigma m_i$ — объем выборки; h — разность между двумя соседними возможными значениями случайной величины $u_i = (x_i - m) / \sqrt{s^2}$ — нормированное значение i-го выборочного значения; $\varphi(u_i)$ — значение плотности вероятности стандартного гауссовского распределения (см. табл. П.1).

При справедливости гипотезы H_0 статистика t при $n \to \infty$ имеет распределение, близкое к χ^2-распределению с $\nu = M - 3$ степенями свободы (здесь учитывается, что по выборке также оценивается два параметра распределения: μ и σ^2).

Нулевая гипотеза H_0 принимается, если

$$t < \chi^2_{\nu;\,1-\alpha},$$

в противном случае гипотезу H_0 отвергают.

Если гипотеза H_0 принимается, то в качестве параметров μ и σ^2 распределения случайной величины X

принимают выборочное среднее значение m и дисперсию s^2.

Следует иметь в виду, что при разбиении выборки на классы должно выполняться условие $m_i' \geqslant 5$, что может быть достигнуто объединением соседних малочисленных классов. При этом в качестве M при определении числа степеней свободы ν следует принимать число групп выборки, оставшихся после объединения частот.

Для проверки гипотезы служат программы 6.15:

Программа 6.15/21

Р3	0	Р2	F3	С/П	×	↑	С/П	F√	Р3	÷	Р4
С/П	Р5	С/П	Р6	С/П	Р7	С/П	↑	F5	—	↑	F3
÷	С/П	↑	F4	×	/—/	↑	F7	+	Fx^2	÷	F1/x
/—/	↑	F2	+	Р2	С/П	БП	×	F2	↑	F6	—
РХ⩽0	F—	0	С/П	1	С/П						

Программа 6.15/34

П0	0	П2	ИП0	С/П	×	С/П	F√	П1	÷
П0	С/П	П3	С/П	П4	С/П	П5	С/П	ИП3	—
ИП1	÷	С/П	ИП0	×	↑	ИП5	—	Fx^2	÷
F1/x	ИП2	+	П2	БП	15	ИП2	ИП4	—	FX<0
43	0	С/П	1	С/П					

Инструкция к программам 6.15: В/О; набрать n С/П; h С/П; s^2 С/П; m С/П; $\chi^2_{\nu;1-\alpha}$ С/П; последовательно набирать m_i С/П; x_i С/П; при этом в РХ появляется значение u_i; $\varphi(u_i)$ С/П ($i=1, \ldots$); после ввода всех данных набрать БП Р Сx (БП 34) С/П; если содержимое регистра РХ $<$РХ$>=0$, то принимается гипотеза H_0, если $<$РХ$>=1$ — гипотеза H_1, в Р2 содержится значение статистики t. В скобках записан адрес перехода для Б3-34.

2. Пусть эмпирическое распределение задано в виде последовательности интервалов $[x_i, x_{i+1}]$ и соответствующих им частот m_i (m_i — сумма частот, которые попали в интервал i).

Интервалы выбраны так, чтобы в каждом было не менее пяти элементов. Число интервалов M, как отмечалось в § 6.1, обычно достигает 8—12. Общий объем выборки $n=\Sigma m_i$ должен быть достаточно большим ($n \geqslant 50$).

В этом случае также используется статистика t и соответствующее решающее правило, описанное выше. Отличие состоит в способе определения теоретической частоты m_i'.

Для определения m_i' по эмпирической выборке находят выборочное среднее m и выборочную дисперсию s^2, используя программы обработки сгруппированных данных (см. § 6.2). Далее переходят к нормированным значениям границ интервалов:

$$u_i=(x_i-m)/\sqrt{s^2}; \quad u_{i+1}=(x_{i+1}-m)/\sqrt{s^2},$$

причем нижнюю границу первого ($i=1$) интервала полагают равной $-\infty$ ($u_1=-\infty$), а верхнюю границу последнего ($i=M$) интервала равной $+\infty$ ($u_{M+1}=+\infty$). Теоретические частоты определяются с использованием функции стандартного гауссовского распределения $\Phi(u_i)$: $m_i'=nP_i$, где $P_i=\Phi(u_{i+1})-\Phi(u_i)$.

Значения функции распределения $\Phi(x)$ приведены в табл. П.2. Более подробные таблицы даны в работе [19]. Далее вычисляется статистика t и принимается решение.

Для проверки гипотезы используются программы в 6.16:

Программа 6.16/21

Р3	0	Р2	Р7	С/П	Р4	С/П	F√	Р5	С/П	Р6	С/П
↑	F4	—	↑	F5	÷	С/П	↑	F7	—	XY	Р7
F3	×	Р8	С/П	↑	F8	—	Fx^2	XY	РНОП	÷	↑
F2	+	Р2	С/П	БП	Р2	F2	↑	F6	—	РХ<0	Р—
0	С/П	1	С/П								

Программа 6.16/34

П0	0	П1	П2	С/П	П3	С/П	F√	П4	С/П
П5	С/П	ИП3	—	ИП4	÷	С/П	↑	ИП1	—
XY	П1	XY	ИП0	×	П6	С/П	↑	ИП6	—
Fx^2	XY	÷	ИП2	+	П2	БП	11	ИП2	ИП5
—	Fx<0	45	0	С/П	1	С/П			

Инструкция к программам 6.16: В/О; набрать n С/П; m С/П; s^2 С/П; $\chi^2_{\nu;1-\alpha}$ С/П; набрать x_{i+1} С/П; при этом в РХ заносится рассчитанное значение u_{i+1}; набрать $\Phi(u_{i+1})$ С/П; m_i С/П; ($i=1, \ldots$); после ввода всех данных набрать БП Р7 (38) С/П; если $<$РХ$>=0$, то принимается гипотеза H_0, если $<$РХ$>=1$, H_0 отвергается (в скобках указан адрес перехода для Б3-34), в Р2 содержится значение статистики t.

Проверка гипотезы о равенстве функций распределения двух случайных величин (критерий знаков). Пусть $\{x_i\}$ и $\{y_i\}$ — две связанные попарно выборки объема n случайных величин X и Y, где $i=1, ..., w$. Нулевая гипотеза состоит в равенстве функций распределения случайных величин X и Y $(H_0: F_X=F_Y)$.

При справедливости нулевой гипотезы для случайной величины $D=X-Y$, очевидно, выполняются равенства:

$$P(D>0)=P(D<0)=0{,}5.$$

Поэтому в качестве нулевой гипотезы можно рассматривать гипотезу

$$H_0': P(D>0)=0{,}5,$$

которая эквивалентна исходной нулевой гипотезе $H_0: F_X=F_Y$.

При проверке гипотезы H_0' при уровне значимости α используется статистика Z_n^+ которая представляет собой число положительных разностей d_i пар значений выборок $\{x_i\}$ и $\{y_i\}$: $d_i=x_i-y_i$ $(i=1, ..., n)$.

В зависимости от постановки задачи и альтернативных гипотез (исходной H_1 и эквивалентной H_1') используются следующие критерии [20]:

а) двухсторонний критерий $(H_1: F_X\neq F_Y;$ $H_1': P(D>0)\neq 0{,}5)$. Нулевые гипотезы H_0 и H_0' принимаются, если

$$Z_{n\alpha/2}<Z_n^+<n-Z_{n\alpha/2},$$

в противном случае они отклоняются;

б) односторонний (правосторонний) критерий $(H_1: F_X<F_Y;$ $H_0': P(D>0)>0{,}5)$. Нулевые гипотезы принимаются, если

$$Z_n^+<n-Z_{n\alpha},$$

в противном случае более обоснованными считаются альтернативные гипотезы;

в) односторонний (левосторонний) критерий $(H_1: F_X>F_Y;$ $H_1': P(D>0)<0{,}5)$. Нулевые гипотезы принимаются, если

$$Z_n^+>Z_{n\alpha},$$

в противном случае они отклоняются.

Здесь $Z_{n\alpha}$ — критические значения, приведенные в табл. П.3.

Для проверки гипотезы служат программы 6.17:

Программа 6.17/21

↑	С/П	—	РХ≥0	FXY	F2	↑	1	+	Р2	F3	↑
1	+	Р3	С/П	БП	Р0	F2	↑	F4	—	РХ≥0	/—/
F3	↑	F4	—	↑	F2	—	РХ≥0	/—/	БП	F6	1
Р6	F2	↑	F5	—	РХ≥0	7	БП	Cx	1	Р8	F3
↑	F5	—	↑	F2	—	РХ≥0	F+	0	С/П	1	С/П

Программа 6.17/34

П0	0	П4	П5	ИП0	С/П	—	FX≥0	10	КИП4
КИП5	С/П	БП	05	ИП4	ИП1	—	FX≥0	30	ИП5
ИП1	—	ИП4	—	FX≥0	30	0	П0	БП	32
1	П0	ИП5	ИП2	—	ИП4	—	FX≥0	43	0
П7	БП	45	1	П7	ИП4	ИП2	—	FX≥0	52
	С/П	1	С/П						

Инструкция к программам 6.17: В/О; 0 Р2 Р3 Р6 Р8; поочередно попарно набирать x_i С/П; y_i С/П $(i=1, ...)$; при этом в Р2 (Р4) содержится Z_i^+; в Р5 (Р3) — i; после ввода всех данных набрать БП Р3 (14); $Z_{n\alpha/2}$ Р4 (Р1); $Z_{n\alpha}$ Р5 (Р2); С/П; <Р6 (Р0)> — индикатор двухстороннего критерия (если на индикаторе 0, то принимается гипотеза H_0, если на индикаторе 1, то — H_1), <Р8 (РХ)> — индикатор левостороннего критерия, <РХ (Р7)> — индикатор правостороннего критерия (в скобках указаны адрес перехода и регистры памяти для Б3-34). Очистка регистров перед началом работы необходима только для Б3-21.

Проверка гипотезы о вероятности появления события в отдельном испытании. Некоторое событие может произойти с вероятностью p, значение которой неизвестно.

Нулевая гипотеза H_0 состоит в том, что неизвестная вероятность p равна гипотетической вероятности p_0: $H_0: p=p_0$.

Требуется при заданном уровне значимости α проверить нулевую гипотезу H_0 по результатам достаточно большого числа испытаний n $(n\geqslant 50—100)$, в ходе которых определена наблюдаемая относительная частота появления события w.

Для проверки гипотезы используется статистика

$$t=(w-p_0)\sqrt{\frac{n}{p_0(1-p_0)}}.$$

При условии справедливости гипотезы H_0 статистика распределена асимптотически (при $n \to \infty$) по гасуссовскому закону (N (0,1)).

В зависимости от постановки задачи и выбора альтернативной гипотезы H_1 используются следующие решающие правила:

а) двухсторонний критерий ($H_1: p \neq p_0$). Нулевая гипотеза H_0 принимается, если $|t| < \lambda_{1-\alpha/2}$, в противном случае она отклоняется;

б) односторонний (правосторонний) критерий ($H_0: p > p_0$). Нулевая гипотеза H_0 принимается, если $t < \lambda_{1-\alpha}$, в противном случае гипотеза H_0 отвергается;

в) односторонний (левосторонний) критерий ($H_1: p < < p_0$). Гипотеза H_0 принимается, если $t > \lambda_{1-\alpha}$, в противном случае нулевая гипотеза отвергается.

Здесь λ_q — квантиль порядка q стандартного гауссовского распределения (см. табл. П.4).

Для проверки гипотезы используются программы 6.18:

Программа 6.18/21

P2	С/П	P3	1	—	/—/	×	↑	F2	÷	F1/x	F√
P4	С/П	P5	С/П	P6	С/П	P7	F5	↑	F3	—	↑
F4	×	P4	Fx²	F√	↑	F6	—	PX≥0	F6	1	БП
PВП	0	P6	F4	F↑	F7	+	P8	F7	—	PX≥0	8
1	БП	—	0	P7	F8	PX≥0	F+	0	С/П	1	С/П

Программа 6.18/34

С/П	П0	1	XY	—	÷	ИП0	÷	F√	С/П
ИП0	—	×	П0	FX<0	27	/—/	С/П	—	FX<0
25	0	П1	БП	27	1	П1	С/П	П3	ИП0
—	FX≥0	37	0	П2	БП	39	1	П2	ИП0
ИП3	+	FX≥0	46	0	С/П	1	С/П		

Инструкция к программам 6.18: В/О; набрать n С/П; p_0 С/П; w С/П; $\lambda_{1-\alpha/2}$ С/П; $\lambda_{1-\alpha}$ С/П; в Р4 (Р0) содержится значение статистики t, в Р6 (Р1) — индикатор двухстороннего критерия, в Р7 (Р2) — индикатор правостороннего критерия, в РХ — индикатор левостороннего критерия (в скобках номера регистров для Б3-34).

Проверка гипотезы о среднем значении гауссовской случайной величины. Случай известной дисперсии. Пусть X имеет распределение N (μ, σ_0^2), где

μ — неизвестное среднее значение, σ_0^2 — известная дисперсия.

Проверяется нулевая гипотеза $H_0: \mu = \mu_0$, где μ_0 — гипотетическое (предполагаемое) среднее значение.

В зависимости от содержательной постановки задачи могут использоваться три альтернативные гипотезы H_1: а) $H_1: \mu \neq \mu_0$; б) $H_1: \mu > \mu_0$; в) $H_1: \mu < \mu_0$.

Нулевая гипотеза проверяется при заданном уровне значимости α — вероятности ошибочного принятия альтернативной гипотезы H_1 при справедливости нулевой гипотезы H_0. Проверка производится по выборке $\{x_i\} = = (x_1, ..., x_n)$ объема n с помощью статистики

$$t = \frac{m - \mu_0}{\sigma_0} \sqrt{n},$$

где m — выборочное среднее.

Статистика t при справедливости гипотезы H_0 имеет гауссовское распределение N (0; 1). В зависимости от вида альтернативной гипотезы H_1 используют следующие критерии:

а) двухсторонний критерий ($H_1: \mu \neq \mu_0$). Гипотеза принимается, если $|t| \leqslant \lambda_{1-\alpha/2}$, в противном случае гипотеза H_0 отвергается;

б) односторонний (правосторонний) критерий ($H_1: : \mu > \mu_0$). Гипотеза H_0 принимается, если $t < \lambda_{1-\alpha}$, в противном случае принимается H_1;

в) односторонний (левосторонний) критерий ($H_1: : \mu < \mu_0$). Гипотеза H_0 принимается, если $t > -\lambda_{1-\alpha}$, в противном случае принимается H_1.

Здесь λ_q — квантиль гауссовского распределения порядка q (см. табл. П.4). Если требуемое значение q в таблице отсутствует, то соответствующее значение λ_q получается с использованием программ интерполяции.

Программы 6.19 применяются, если выборочное среднее предварительно вычислено, например, с использованием программ 6.2;

Программа 6.19/21

P2	С/П	P3	С/П	P4	С/П	P5	С/П	P6	С/П	↑	F2
—	↑	F3	÷	↑	F6	F√	×	P6	↑	F5	—
PX≥0	P5	1	P7	БП	P/—/	0	P7	F6	↑	F5	+
PX<0	P7	1	P8	БП	PCx	0	P8	F6	Fx²	F√	↑
F4	—	PX≥0	P9	1	С/П	0	С/П				

Программа 6.19/34

П6	С/П	П1	С/П	П2	С/П	П3	С/П	П4	С/П
П6	ИП0	—	ИП1	÷	ИП4	F√	×	П7	ИП2
—	FX≥0	27	1	П8	БП	29	0	П8	ИП7
ИП2	+	FX≥0	38	0	П9	БП	40	1	П9
ИП7	FX≥0	45	БП	46	/—/	ИП3	—	FX≥0	52
1	С/П	0	С/П						

Инструкция к программам 6.19: В/О; набрать μ_0 С/П; σ_0 С/П; $\lambda_{1-\alpha/2}$ С/П; $\lambda_{1-\alpha}$ С/П; n С/П; m С/П; после завершения программы значение t содержится в Р6 (Р7), <Р7> (<Р8>) — индикатор правостороннего критерия, <Р8> (<Р9>) — индикатор левостороннего критерия, <РХ> — индикатор двухстороннего критерия (в скобках указаны регистры для Б3-34).

Программы 6.20 предназначены для расчета выборочного среднего m, статистики t и проверки гипотезы непосредственно по выборке $\{x_i\}$:

Программа 6.20/21

Р6	0	Р7	Р8	F6	↑	F8	—	Р6	F7	1	+
Р7	↑	F6	XY	÷	↑	F8	+	Р8	С/П	БП	↑
↑	F2	—	↑	F3	÷	↑	F7	F√	×	Р6	Fx^2
F√	↑	F4	—	РХ≥0	РСx	1	С/П	0	С/П		
F4	—	РХ≥0	РСx	1	Р5	БП	FСx	0	Р5	F6	↑
F4	+	РХ<0	Р9	1	С/П	0	С/П				

Программа 6.20/34

П0	С/П	П1	С/П	П2	С/П	П3	0	П4	П5
С/П	ИП5	+	П5	КИП4	БП	10	ИП5	ИП4	÷
П6	ИП0	—	ИП1	÷	ИП4	F√	×	П7	ИП2
—	FX≥0	37	1	П8	БП	39	0	П8	ИП7
ИП2	+	FX≥0	48	0	П9	БП	50	1	П9
ИП7	FX≥0	55	БП	56	/—/	ИП3	—	FX≥0	62
1	С/П	0	С/П						

Инструкция к программе 6.20/21 (первый вариант программы — проверка двухстороннего критерия): В/О; набрать μ_0 Р2; σ_0 Р3; $\lambda_{1-\alpha/2}$ Р4; далее последовательно набирать x_i С/П; после ввода всех данных набрать команды БП Р4 С/П; <РХ> — индикатор двухстороннего критерия, $<Р7>=n$, $<Р8>=m$, $<Р6>=t$.

Инструкция к программе 6.20/21 (второй вариант программы — проверка односторонних критериев, отличается фрагментом, начинающимся с адреса 55): В/О; набрать μ_0 Р2; σ_0 Р3; $\lambda_{1-\alpha}$ Р4; далее действовать в соответствии с предыдущей инструкцией; <РХ> — индикатор левостороннего критерия, <Р5> — индикатор правостороннего критерия.

Проверку односторонних критериев можно проводить и после проверки двухстороннего критерия. Для этого после выполнения первого варианта программы, не изменяя содержимого регистров памяти, ввести, начиная с адреса 55, заключительный фрагмент второго варианта. Далее набрать $\lambda_{1-\alpha}$ Р4; БП Р4 С/П.

Инструкция к программе 6.20/34: В/О; набрать μ_0 С/П; σ_0 С/П; $\lambda_{1-\alpha}$ С/П; $\lambda_{1-\alpha/2}$ С/П; последовательно набрать x_i С/П; после ввода всех данных набрать БП 17 С/П; содержимое регистров — $<Р4>=n$, $<Р6>=m$, $<Р7>=t$, <Р8> — индикатор правостороннего критерия, <Р9> — левостороннего, <РХ> — двухстороннего.

Контрольный пример. Исходные данные: $\{x_i\}=1, 2, 3, 1, 2, 3$; $\lambda_{1-\alpha/2}=2{,}5758$; $\lambda_{1-\alpha}=2{,}3263$ $(\alpha=0{,}01)$, $\mu_0=3$. Результаты: наблюдаемое значение статистики $t=-0{,}6$; гипотеза H_0 отвергается.

Случай неизвестной дисперсии (t-критерий). Пусть случайная величина X имеет гауссовское распределение $N(\mu, \sigma^2)$, причем параметры μ и σ^2 неизвестны. Проверяется нулевая гипотеза $H_0: \mu=\mu_0$, где μ_0 — гипотетическое среднее значение. Нулевая гипотеза проверяется при заданном уровне значимости α — вероятности ошибочного принятия альтернативной гипотезы H_1 при справедливости нулевой гипотезы H_0. Проверка производится по выборке $\{x_i\}=(x_1,\ldots, x_n)$ объема n с помощью статистики

$$t=\frac{m-\mu_0}{s^2}\sqrt{n},$$

где m — выборочное среднее; s^2 — выборочная дисперсия.

Статистика t при справедливости гипотезы H_0 имеет t-распределение Стьюдента с $\nu=n-1$ степенями свободы.

В зависимости от вида альтернативной гипотезы H_1 применяются следующие критерии:

а) двухсторонний критерий $(H_1: \mu\neq\mu_0)$. Гипотеза H_0 принимается, если $|t|<t_{\nu;\,1-\alpha/2}$, в противном случае H_0 отвергается;

б) односторонний (правосторонний) критерий $(H_1: \mu>\mu_0)$. Гипотеза H_0 принимается, если $t<t_{\nu;\,1-\alpha}$, в противном случае H_0 отвергается;

в) односторонний (левосторонний) критерий ($H_1: \mu<\mu_0$). Гипотеза H_0 принимается, если $t>-t_{\nu;1-\alpha}$, в противном случае принимается H_1.

Здесь $t_{\nu q}$ — квантиль q-го порядка t-распределения Стьюдента с ν степенями свободы (см. табл. П.5).

Если выборочное среднее значение m и дисперсия s^2 определены заранее, например, с использованием программ 6.2, то используют программы 6.19 предыдущего пункта. Отличия заключаются в использовании вместо известной дисперсии σ_0^2 выборочной дисперсии s^2 и вместо квантилей гауссовского распределения λ_q квантилей t-распределения $t_{\nu;q}$.

Для проверки гипотезы непосредственно по выборке $\{x_i\}$ и для одновременного расчета величин m и s^2 используются программы 6.21:

Программа 6.21/21

↑	F2	+	P2	F3	XY	Fx^2	+	P3	F4	1	+
P4	XY	P5	С/П	БП	P0	F2	↑	F4	÷	P2	Fx^2
↑	F4	×	/—/	↑	F3	+	↑	F5	÷	P3	F5
С/П	F2	↑	F6	—	↑	F3	F$\sqrt{}$	÷	↑	F4	F$\sqrt{}$
×	Fx^2	F$\sqrt{}$	↑	F7	—	PX<0	F+	0	С/П	1	С/П
	↑	F7	—	PX≥0	P+	1	С/П	0	С/П		
	↑	F7	+	PX<0	P+	1	С/П	0	С/П		

Программа 6.21/34

П2	С/П	П3	С/П	П6	0	П0	П1	П4	С/П
Fx^2	FBx	ИП0	+	П0	XY	ИП1	+	П1	ИП4
П5	КИП4	С/П	БП	10	ИП0	Fx^2	ИП4	÷	ИП1
XY	—	ИП5	÷	П7	ИП0	ИП4	÷	П8	ИП6
—	ИП4	F$\sqrt{}$	×	ИП7	F$\sqrt{}$	÷	П9	FX<0	51
/—/	ИП3	—	FX≥0	59	1	ПА	БП	61	0
ПА	ИП9	ИП2	—	FX≥0	70	1	ПВ	БП	72
0	ПВ	ИП9	ИП2	+	FX≥0	80	0	БП	81
1	С/П								

Инструкция к программе 6.21/21: В/О; набрать 0 Р2 Р3 Р4, μ_0 Р6 $t_{\nu;1-\alpha/2}$ ($t_{\nu;1-\alpha}$ — для односторонних критериев) Р7; последовательно набирать x_i С/П, при этом

$$<P2>=\sum_{j=1}^{i} x_j; \quad <P3>=\sum_{j=1}^{i} x_j^2, \quad <P4>=i, \quad <PX>=$$

$=<P5>=i-1$; после ввода последнего значения набрать команды БП Р3 С/П, при этом $<P2>=m$, $<P3>=s^2$, $<P4>=n$, $<PX>=<P5>=n-1$. В зависимости от проверяемой альтернативной гипотезы используются различные завершающие фрагменты программы, начинающиеся с адреса 81. После расчета оценок математического ожидания и дисперсии выполнить команды БП F6 С/П; при этом в РХ содержится индикатор двухстороннего или односторонних критериев.

Инструкция к программе 6.21/34: В/О; набрать $t_{\nu;1-\alpha}$ С/П; $t_{\nu;1-\alpha/2}$ С/П; μ_0 С/П; поочередно набирать x_i С/П; после ввода всех данных набрать БП 25 С/П; $<P0>=\Sigma x_i$, $<P1>=\Sigma x_i^2$, $<P4>=n$, $<P7>=s^2$, $<P8>=m$, $<P9>=t$, $<PA>$ — индикатор двухстороннего критерия, $<P8>$ — правостороннего, $<PX>$ — левостороннего.

Контрольный пример. Исходные данные: $\{x_i\}=82$; 72; 80; 72; 70; 82; 74; 78; 76; 78; $\mu_0=80$; $t_{\nu;1-\alpha}=2{,}262$, $t_{\nu;1-\alpha}$ 1,833 ($\alpha=0{,}05$). Результаты: наблюдаемое значение статистики $t=-2{,}6475678$; гипотеза H_0 принимается только при конкурирующей гипотезе $H_1: \mu>\mu_0$, в других случаях она отклоняется.

Проверка гипотезы о равенстве средних значений двух гауссовских случайных величин. Пусть случайные величины X_1 и X_2 имеют распределения $N(\mu_1, \sigma_1^2)$ и $N(\mu_2, \sigma_2^2)$. Дисперсии σ_1^2 и σ_2^2 распределений полагаются известными. Проверяется нулевая гипотеза H_0: $\mu_1=\mu_2$.

Нулевая гипотеза проверяется при заданном уровне значимости α по выборке $\{x_{1i}\}=(x_{11}, x_{12}, \ldots, x_{1n})$ объема n_1 случайной величины X_1 и по выборке $\{x_{2i}\}=(x_{21}, x_{22}, \ldots, x_{2n})$ объема n_2 случайной величины X_2. При этом используется статистика

$$t=\frac{m_1-m_2}{\sqrt{(\sigma_1^2/n_1)+(\sigma_2^2/n_2)}},$$

где m_1 и m_2 — выборочные средние значения выборок $\{x_{1i}\}$ и $\{x_{2i}\}$. При справедливости нулевой гипотезы H_0 статистика t имеет распределение $N(0; 1)$. В зависимости от альтернативной гипотезы H_1 используются следующие критерии:

а) двухсторонний критерий ($H_1: \mu_1\neq\mu_2$). Гипотеза H_0 принимается, если $|t|<\lambda_{1-\alpha/2}$, в противном случае гипотеза H_0 отвергается;

б) односторонний (правосторонний) критерий ($H_1: \mu_1>\mu_2$).

Гипотеза H_0 принимается, если $t<\lambda_{1-\alpha}$, в противном случае гипотеза H_0 отвергается;

в) односторонний (левосторонний) критерий ($H_1: \mu_1<\mu_2$). Гипотеза H_0 принимается, если $t>-\lambda_{1-\alpha}$, в противном случае гипотеза H_0 отвергается.

Здесь λ_q — квантиль q-го порядка гауссовского распределения (см. табл. П.4).

Программы 6.22 используются после определения выборочных средних значений m_1 и m_2, для чего могут применяться программы 6.2:

Программа 6.22/21

P2	С/П	P3	С/П	P→	С/П	↑	С/П	÷	P←	↑	С/П
—	P4	С/П	↑	С/П	÷	↑	P→	+	F√	F1/x	↑
F4	×	P4	Fx²	F√	↑	F2	—	PX≥0	F6	1	БП
PВП	0	P5	F4	↑	F3	+	P8	F3	—	PX≥0	8
1	БП	F—	0	P6	F8	PX≥0	F+	0	С/П	1	С/П

Программа 6.22/34

С/П	—	С/П	С/П	÷	С/П	С/П	÷	+	F√
÷	П0	FX<0	15	/—/	С/П	—	FX≥0	23	1
П1	БП	25	0	П1	С/П	П3	ИП0	+	FX≥0
35	0	П2	БП	37	1	П2	ИП0	ИП3	—
FX<0	44	0	С/П	1	С/П				

Инструкция к программе 6.22/21: В/О; набрать $\lambda_{1-\alpha/2}$ С/П; $\lambda_{1-\alpha}$ С/П; m_1 С/П; σ_1^2С/П; n_1 С/П; m_2 С/П; σ_2^2 С/П; n_2 С/П; <P4>$=t$, <P5> — индикатор двухстороннего критерия, <P6> — правостороннего, <PX> — левостороннего.

Инструкция к программе 6.22/34: В/О; набрать m_1 С/П; m_2 С/П; σ_1^2 С/П; n_1 С/П; σ_2^2 С/П; n_2 С/П; $\lambda_{1-\alpha/2}$ С/П; $\lambda_{1-\alpha}$ С/П; <P0>$=t$, <P1> — индикатор двухстороннего критерия, <P2> — левостороннего, <PX> — правостороннего.

Пусть случайные величины X_1 и X_2 имеют распределения $N(\mu_1, \sigma^2)$ и $N(\mu_2, \sigma^2)$ с равными, но неизвестными дисперсиями. Проверяется нулевая гипотеза $H_0: \mu_1=\mu_2$.

Нулевая гипотеза проверяется при заданном уровне значимости α по выборке $\{x_{1i}\}=(x_{11}, ..., x_{1n})$ объема n_1 величины X_1 и по выборке $\{x_{2i}\}=(x_{21}, ..., x_{2n})$ объема n_2 случайной величины X_2, по которым вычисляются выборочные средние значения m_1 и m_2 и выборочные дисперсии s_1^2 и s_2^2. Далее используется статистика [20]

$$t=\frac{m_1-m_2}{\sqrt{(n_1-1)\,s_1^2+(n_2-1)\,s_2^2}}\sqrt{\frac{n_1 n_2 (n_1+n_2-2)}{n_1+n_2}}.$$

При условии справедливости нулевой гипотезы H_0 статистика t имеет t-распределение Стьюдента с $\nu=n_1+n_2-2$ степенями свободы. В зависимости от альтернативной гипотезы H_1 используются следующие критерии:

а) двухсторонний критерий ($H_1: \mu_1\neq\mu_2$). Гипотеза H_0 принимается, если $|t|<t_{\nu;\,1-\alpha/2}$, в противном случае H_0 отвергается;

б) односторонний (правосторонний) критерий ($H_1: \mu_1>\mu_2$). Гипотеза H_0 принимается, если $t<t_{\nu;1-\alpha}$, в противном случае принимается альтернативная гипотеза H_1;

в) односторонний (левосторонний) критерий ($H_1: \mu_1<\mu_2$). Гипотеза H_0 принимается, если $t>-t_{\nu;\,1-\alpha}$, в противном случае принимается гипотеза H_1.

Здесь $t_{\nu q}$ — квантиль порядка q t-распределения Стьюдента с ν степенями свободы (см. табл. П.5). Программы 6.23 используются после расчета выборочных средних значений m_1 и m_2 и выборочных дисперсий s_1^2 и s_2^2, которые могут быть получены с помощью программы 6.2:

Программа 6.23/21

F7	↑	F4	+	2	—	÷	↑	F4	÷	↑	F7
÷	P→	F4	1	—	↑	F3	×	P→	F7	1	—
↑	F6	×	↑	P←	+	↑	P←	×	F1/x	F√	P→
F2	↑	F5	—	↑	P←	×	P2	Fx²	F√	↑	F8
—	PX≥0	—	1	С/П	0	С/П					
								↑	F8	—	PX≥0
8	1	С/П	0	С/П							
								↑	F8	+	PX≥0
8	0	С/П	1	С/П							

Программа 6.23/34

П0	С/П	П1	С/П	П2	С/П	П3	С/П	П4	С/П
П5	ИП2	+	2	—	ИП2	×	ИП5	×	ИП2
ИП5	+	÷	F√	ИП0	ИП3	—	×	ИП2	1
—	ИП1	×	ИП5	1	—	ИП4	×	+	F√
÷	П6	С/П	↑	ИП6	Fx²	F√	—	FX≥0	54
0	П7	БП	56	1	П7	С/П	П9	↑	ИП6
—	FX≥0	67	0	П8	БП	69	1	П8	ИП6
ИП9	/—/	—	FX≥0	78	0	П9	С/П	1	П9
С/П									

Инструкция к программе 6.23/21: В/О; набрать m_1 Р2; s_1^2 Р3; n_1 Р4; m_2 Р5; s_2^2 Р6; n_2 Р7; $t_{\nu;\,1-\alpha/2}$ ($t_{\nu;\,1-\alpha}$ — в одностороннем случае) Р8 С/П. В зависимости от критерия используются различные завершающие фрагменты программы, начиная с адреса 72 (для двухстороннего, правостороннего и левостороннего критериев). В Р2 содержится t, в РХ — индикатор соответствующего критерия.

Инструкция к программе 6.23/34: В/О; набрать m_1 С/П; s_1^2 С/П; n_1 С/П; m_2 С/П; s_2^2 С/П; n_2 С/П; $t_{\nu;\,1-\alpha/2}$ С/П; $t_{\nu;1-\alpha}$ С/П; $<$Р9$>$ — индикатор левостороннего критерия, $<$Р7$>$ — двухстороннего, $<$Р8$>$ — правостороннего; $<$Р6$>=t$.

Контрольный пример. Исходные данные: $m_1=2{,}43$; $s^1_2=16{,}4$; $n_1=14$; $m_2=4{,}9$; $s_2^2=22{,}5$; $n_2=10$; $t_{\nu;\,1-\alpha/2}=$ $=2{,}074$; $t_{\nu;\,1-\alpha}=1{,}717$ ($\nu=22$; $\alpha=0{,}05$). Результаты: Вычисленное значение статистики $t=-1{,}372387$; гипотеза H_0 принимается при всех возможных альтернативных гипотезах.

Пусть случайные величины X_1 и X_2 имеют распределения $N(\mu_1, \sigma_1^2)$ и $N(\mu_2, \sigma_2^2)$, причем известно, что $\sigma_1^2 \neq$ $\neq \sigma_2^2$, либо гипотеза $\sigma_1^2 \neq \sigma_2^2$ при проверке ее с использованием F-критерия (см. ниже) была отклонена.

Нулевая гипотеза H_0: $\mu_1=\mu_2$.

Проверка гипотезы при уровне значимости α производится по двум независимым выборкам $\{x_{1i}\}$ и $\{x_{2i}\}$ объема n_1 и n_2, по которым определяются выборочные средние m_1 и m_2 и выборочные дисперсии s_1^2 и s_2^2 соответственно. Используется статистика

$$t=\frac{m_1-m_2}{\sqrt{s_1^2/n_1+s_2^2/n_2}}.$$

При справедливости нулевой гипотезы H_0 статистика t имеет t-распределение Стьюдента.

В зависимости от постановки задачи рассматриваются различные альтернативные гипотезы H_1 и соответствующие критерии:

а) двухсторонний критерий ($H_1:\mu_1\neq\mu_2$). Гипотеза принимается, если $|t|<t_{\nu;\,1-\alpha/2}$, в противном случае гипотеза H_0 отклоняется;

б) односторонний (правосторонний) критерий ($H_1:\mu_1>\mu_2$). Гипотеза H_0 принимается, если $t<t_{\nu;\,1-\alpha}$, в противном случае гипотеза H_0 отвергается;

в) односторонний (правосторонний) критерий (H_1 : $\mu_1<\mu_2$). Гипотеза H_0 принимается, если $t<-t_{\nu;\,1-\alpha}$, в противном случае следует принять гипотезу H_1.

Здесь $t_{\nu\,q}$ — квантиль q-го порядка t-распределения Стьюдента с ν степенями свободы (см. табл. П.5). Величина ν — наибольшее целое число, не превосходящее величины ν' [6]; $\nu=\{\nu'\}$, где

$$\nu'=\frac{(s_1^2/n_1+s_2^2/n_2)^2}{\dfrac{(s_1^2/n_1)^2}{n_1-1}+\dfrac{(s_2^2/n_2)^2}{n_1-1}}.$$

Величина ν лежит в интервале между наименьшим из чисел (n_1-1) и (n_2-1) и их суммой (n_1+n_2-2) [6]. Программа 6.24 используется после расчета по выборкам $\{x_{1i}\}$ и $\{x_{2i}\}$ выборочных средних и дисперсий:

Программа 6.24/21

F2	↑	F5	—	P2	F3	↑	F4	÷	P3	F6	↑
F7	÷	P6	↑	F3	+	P8	F√	F1/x	↑	F2	×
P2	F4	1	—	↑	F3	Fx²	÷	F1/x	P3	F7	1
—	↑	F6	Fx²	÷	F1/x	↑	F3	+	↑	F8	Fx²
÷	F1/x	С/П	↑	F2	Fx²	F√	—	PX≥0	+	0	С/П
			↑	F2	—	PX≥0	F+	0	С/П	1	С/П
			↑	F2	+	PX≥0	F+	0	С/П	1	С/П

Программа 6.24/34

С/П	—	С/П	С/П	П0	1	—	П1	1	+
÷	П2	С/П	С/П	П3	÷	П4	+	П5	F√
÷	П6	ИП3	1	—	П7	ИП5	Fx²	ИП2	Fx²
ИП1	÷	ИП4	Fx²	ИП7	÷	+	÷	П8	КИП8
ИП8	С/П	П0	С/П	ИП6	FX<0	48	/—/	—	FX≥0
55	0	П1	БП	57	1	П1	ИП0	ИП6	—
FX≥0	66	0	П2	БП	68	1	П2	ИП6	ИП0
+	FX≥0	75	0	С/П	1	С/П			

Инструкция к программе 6.24/21: В/О; набрать m_1 Р2; s_1^2 Р3; n_1 Р4; m_2 Р5; s_2^2 Р6; n_2 Р7 С/П; при этом в РХ содержится ν', в Р2 — t; набрать $t_{\nu;1-\alpha/2}$ С/П; $<$РХ$>$ — индикатор двухстороннего критерия. Для проверки правостороннего или левостороннего критериев используются соответствующие фрагменты программ, начинающиеся с адреса 83, при этом вместо $t_{\nu;1-\alpha/2}$ вводится значение $t_{\nu;1-\alpha}$, <РХ> — индикатор критерия: если <РХ> $\neq 0$, то гипотеза отвергается; если <РХ>$=0$ — принимается.

Инструкция к программе 6.24/34: В/О; набрать m_1 С/П; m_2 С/П; s_1^2 С/П; n_1 С/П; s_2^2 С/П; n_2 С/П; $t_{\nu;1-\alpha}$ С/П; $t_{\nu;1-\alpha/2}$ С/П; <РХ> — индикатор левостороннего критерия, <Р1> — двустороннего, <Р2> — правостороннего, <Р6>$=t$, <Р8>$=\nu$.

Пусть случайные величины X_1 и X_2 имеют распределения $N(\mu_1, \sigma_1^2)$ и $N(\mu_2, \sigma_2^2)$, причем дисперсии $\sigma_н^2$ и σ_2^2 неизвестны. Получены две зависимые выборки каждой из этих случайных величин $\{x_{1i}\}$ и $\{x_{2i}\}$ одинакового объема n. Например, двумя приборами в одном и том же порядке производятся измерения деталей (массы, размеров и т. д.).

Нулевая гипотеза $H_0 : \mu_1 = \mu_2$. Альтернативная гипотеза $H_1 : \mu_1 \neq \mu_0$. Проверка гипотезы производится с использованием статистики

$$t = \frac{\bar{d}\sqrt{n}}{s_d},$$

где $\bar{d} = (\Sigma d_i)/n$ — средняя разность элементов выборок $\{x_{1i}\}$ и $\{x_{2i}\}$ с одинаковыми номерами; $s_d = \sqrt{\dfrac{\Sigma d_i^2 - (\Sigma d_i)^2/n}{n-1}}$ — выборочное среднее квадратическое значение разностей элементов выборок с одинаковыми номерами.

Здесь $d_i = x_{1i} - x_{2i}$ $(i = 1, ..., n)$ — разности элементов выборок $\{x_{1i}\}$ и $\{x_{2i}\}$.

При справедливости гипотезы H_0 статистика t имеет t-распределение Стьюдента с $\nu = n - 1$ степенями свободы [19].

Нулевая гипотеза H_0 принимается при уровне значимости α, если $|t| < t_{\nu;1-\alpha/2}$, в противном случае гипотеза H_0 отклоняется.

Здесь $t_{\nu;q}$ — квантиль q-го порядка t-распределения Стьюдента с ν степенями свободы (см. табл. П.5).

Программы 6.25 позволяют по выборкам $\{x_{1i}\}$ и $\{x_{2i}\}$ определить значение статистики и принять решение относительно гипотезы:

Программа 6.25/21

↑	С/П	—	↑	F2	+	Р2	F3	XY	Fx^2	+	Р3
F4	Р5	1	+	Р4	С/П	БП	Р0	Р6	F2	↑	F4
÷	Р7	Fx^2	↑	F4	×	/—/	↑	F3	+	↑	F5
÷	Р8	F1/x	↑	F4	×	F√	↑	F7	×	Р5	Fx^2
F√	↑	F6	—	РХ⩾0	Р+	1	С/П	0	С/П		

Программа 6.25/34

0	П0	П1	П4	С/П	С/П	—	↑	Fx^2	ИП1
+	П1	XY	ИП0	+	П0	КИП4	БП	04	ИП0
ИП4	÷	П2	Fx^2	ИП4	×	ИП1	XY	—	ИП4
1	—	÷	П3	F√	ИП2	XY	÷	ИП4	F√
×	П5	FX<0	45	/—/	С/П	—	FX⩾0	51	1
С/П	0	С/П							

Инструкция к программе 6.25/21: В/О; набрать 0 Р2 Р3 Р4; попарно набирать i-е реализации первой и второй выборок x_{1i} С/П; x_{2i} С/П; при этом после ввода x_{2i} в РХ содержится значение i; после ввода всех данных набрать $t_{\nu;1-\alpha/2}$ БП Р÷С/П; <РХ> — индикатор критерия, <Р2>$=\Sigma d_i$; <Р3>$=\Sigma d_i^2$; <Р4>$=n$; <Р5>$=t$, <Р7>$=\bar{d}$, <Р8>$=s_d^2$.

Инструкция к программе 6.25/34: В/О; попарно набирать x_{1i} С/П; x_{2i} СП; после ввода всех данных выполнить команды БП 19 С/П; набрать $t_{\nu;1-\alpha/2}$ С/П; <Р0>$=\Sigma d_i$, <Р1>$=\Sigma d_i^2$, <Р2>$=\bar{d}$, <Р3>$=s_i^2$, <Р4>$=n$, <Р5>$=t$, <РХ> — индикатор критерия.

Контрольный пример. Исходные данные: $\{x_{1i}\}$=2; 3; 5; 6; 8; 10; $\{x_{2i}\}$=10; 3; 6; 1; 7; 4; $t_{5;1-\alpha/2} = 2{,}571$ $(\alpha = 0{,}05)$. Результаты: вычисленное значение $t = 2{,}4446054 \cdot 10^{-1}$, гипотеза H_0 принимается.

Очевидным образом процедура проверки гипотезы H_0 может быть преобразована и для других альтернативных гипотез H_1 $(\mu_1 > \mu_2$ и $\mu_1 < \mu_2)$.

Проверка гипотезы о дисперсии гауссовской случайной величины. Пусть случайная величина X имеет распределение $N(\mu, \sigma^2)$, где μ и σ^2 — неизвестные среднее и дисперсия. Нулевая гипотеза H_0: $\sigma^2 = \sigma_0^2$, где σ_0^2 — гипотетическая (предполагаемая) дисперсия.

Проверка гипотезы H_0 при уровне значимости α производится по результатам выборки $\{x_i\}$ объема n с использованием статистики

$$t = (n-1)\, s^2/\sigma_0^2,$$

где s^2 — выборочная дисперсия.

При справедливости нулевой гипотезы статистика t имеет χ^2-распределение с $\nu=n-1$ степенями свободы.

В зависимости от альтернативной гипотезы H_1 используют следующие критерии:

а) двухсторонний критерий ($H_1:\sigma^2\neq\sigma_0^2$). Гипотеза принимается, если

$$\chi^2_{\nu;\,\alpha/2}\leqslant t\leqslant\chi^2_{\nu;\,1-\alpha/2},$$

в противном случае гипотеза H_0 отклоняется;

б) односторонний (правосторонний) критерий (H_1: $\sigma^2>\sigma_0^2$). Гипотеза H_0 принимается, если $t<\chi^2_{\nu;\,1-\alpha}$, в противном случае H_0 отвергается;

в) односторонний (левосторонний) критерий ($H_1:\sigma^2<\sigma_0^2$). Гипотеза H_0 принимается, если $t>-\chi^2_{\nu\alpha}$, в противном случае гипотеза H_0 отклоняется.

Здесь $\chi^2_{\nu q}$ — квантиль q-го порядка χ^2-распределения с $\nu=n-1$ степенями свободы (см. табл. П.6). Для больших ν ($\nu>100$) квантили χ^2-распределения могут быть рассчитаны с помощью программ 6.26, использующих разложение Корниша—Фишера [20]:

$$\chi^2_{\nu q}\approx\nu+\lambda_q\sqrt{2\nu}+\frac{2}{3}(\lambda_q^2-1)+\frac{1}{9\sqrt{2\nu}}(\lambda_q^3-7\lambda_q),$$

где λ_q — квантиль q-го порядка гауссовского стандартного распределения (см. табл. П.1).

Программа 6.26/21

Р2	2	×	F√	Р3	С/П	Р4	↑	×	Р5	×	Р6
F4	7	×	/—/	↑	F6	+	9	÷	↑	F3	÷
Р6	F5	1	—	↑	2	×	3	÷	Р5	F4	↑
F3	×	↑	F2	+	↑	F5	+	↑	F6	+	Р3
С/П	БП	Р1									

Программа 6.26/34

П0	С/П	П1	2	×	F√	П2	×	ИП1	+
ИП0	Fx^2	П3	1	—	2	×	3	÷	+
ИП0	ИП3	×	ИП0	7	×	—	9	÷	ИП2
÷	+	С/П							

Инструкция к программе 6.26/21: В/О; набрать ν С/П; λ_q С/П; в РХ и Р3 значение $\chi^2_{\nu q}$; при необходимости

расчета нового значения $\chi^2_{\nu q'}$, при том же значении ν набрать $\lambda_{q'}$ С/П.

Инструкция к программе 6.26/34: В/О; набрать λ_q С/П; ν С/П; <РХ>$=\chi^2_{\nu q}$.

Программы 6.27 используются в случае, когда выборочная дисперсия s^2 рассчитана заранее, например, с использованием программ 6.2:

Программа 6.27/21

Р2	С/П	Р3	С/П	↑	1	—	Р4	С/П	Р5	С/П	Р6
С/П	Р7	С/П	Р8	РНОП	F2	↑	F3	÷	↑	F4	×
Р2	↑	F5	—	РХ⩾0	F6	F6	—	РХ<0	F6	0	БП
РВП	1	Р3	F2	↑	F7	—	РХ<0	Р8	0	БП	F8
1	Р4	F2	↑	F8	—	РХ⩾0	F+	0	С/П	1	С/П

Программа 6.27/34

С/П	↑	1	—	С/П	÷	×	↑	↑	П0
С/П	П1	С/П	П2	С/П	П3	С/П	—	FX⩾0	29
ИП3	XY	—	FX⩾0	29	0	П4	БП	31	1
П4	ИП0	ИП1	—	FX⩾0	40	0	П5	БП	42
1	П5	ИП3	XY	—	FX⩾0	49	0	С/П	1
С/П									

Инструкция к программе 6.27/21: В/О; набрать s^2 С/П; σ_0^2 С/П; n С/П; $\chi^2_{\nu;\alpha/2}$ С/П; $\chi^2_{\nu;\,1-\alpha/2}$ С/П; $\chi^2_{\nu;1-\alpha}$ С/П; $\chi^2_{\nu\alpha}$ С/П; <Р2>$=t$, <Р3> — индикатор двустороннего критерия, <Р4>-правостороннего, <РХ> — левостороннего.

Инструкция к программе 6.27/34: В/О; набрать s^2 С/П; n С/П; σ_0^2 С/П; $\chi^2_{\nu\alpha}$ С/П; $\chi^2_{\nu;\,1-\alpha}$ С/П; $\chi^2_{\nu;\,1-\alpha/2}$ С/П; $\chi^2_{\nu;\,\alpha/2}$ С/П; <Р0>$=t$, <Р4> — индикатор двухстороннего критерия, <Р5> — левостороннего, <РХ> — правостороннего.

Программы 6.28 используют непосредственно выборку $\{x_i\}$.

Программа 6.28/21

↑	F3	+	Р3	F4	XY	Fx^2	+	Р4	F5	Р2	1
+	Р5	С/П	БП	Р0	F3	Fx^2	↑	F5	÷	↑	F4
XY	—	↑	F2	÷	Р3	↑	F2	×	Р2	С/П	F2
↑	F4	÷	↑	Р2	F5	—	РХ⩾0	F—	F6	—	РХ<0
F—	0	БП	—	1	Р7	С/П					
							РХ<0	Р8	0	БП	F8
1	Р7	F2	↑	F6	—	РХ⩾0	F—	0	С/П	1	С/П

Программа 6.28/34

П2	С/П	П3	С/П	П5	С/П	П6	С/П	П7	0
П0	П1	П4	С/П	↑	Fx^2	ИП0	+	П0	XY
ИП1	+	П1	КИП4	БП	13	ИП1	Fx^2	ИП4	÷
ИП0	XY	—	ИП7	÷	П8	ИП7	ИП4	1	—
÷	×	ПВ	ИП1	ИП4	÷	ПС	ИП8	↑	↑
ИП5	—	FX⩾0	63	ИП6	XY	—	FX⩾0	63	0
П9	БП	65	1	П9	ИП8	ИП2	—	FX⩾0	74
0	ПА	БП	76	1	ПА	ИП3	XY	—	FX⩾0
83	0	С/П	1	С/П					

Инструкция к программе 6.28/21: В/О; 0 Р3 Р4 Р5; поочередно набирать значения x_i С/П; после ввода всех данных выполнить команды БП С/П; при этом в Р3 содержится s^2; набрать σ_0^2 Р4; $\chi^2_{\nu;\,\alpha/2}$ Р5; $\chi^2_{\nu;\,1-\alpha/2}$ Р6 С/П; $<$РХ$>$ = $<$Р7$>$ — индикатор двухстороннего критерия. Для проверки односторонних критериев начиная с адреса 71 вводится другой завершающий фрагмент программы, а в Р5 и Р6 заносятся значения $\chi^2_{\nu\alpha}$ и $\chi^2_{\nu;\,\alpha-1}$, при этом в РХ — индикатор левостороннего критерия, в Р7 — правостороннего.

Инструкция к программе 6.28/34: В/О; набрать $\chi^2_{\nu\alpha}$ С/П; $\chi^2_{\nu;\,1-\alpha}$ С/П; $\chi^2_{\nu;\,\alpha/2}$ С/П; $\chi^2_{\nu;\,1-\alpha/2}$ С/П; σ_0^2С/П; поочередно набирать значения x_i С/П; после ввода всех данных выполнить команды БП 26 С/П; $<$РХ$>$ — индикатор правостороннего критерия, $<$РА$>$ — левостороннего, $<$Р9$>$ — двухстороннего, $<$Р8$>$ $=t$, $<$Р4$>$ $=n$, $<$РВ$>$ $=s^2$, $<$РС$>$ $=m$. При необходимости обработки нового массива $\{x_i\}$ $(i=1, ..., n)$ при неизменных остальных величинах выполнить команду БП 09 и вводить данные x_i С/П.

Проверка гипотезы о равенстве дисперсий двух гауссовских случайных величин (F-критерий). Пусть случайные величины X_1 и X_2 имеют распределения N_2 (μ_1, σ_1^2) 0 и N (μ_2, σ_2^2). Нулевая гипотеза H_0: $\sigma_1^2=\sigma_2^2$. Она проверяется при заданном уровне значимости α по результатам двух независимых выборок: $\{x_{1i}\}$ объема n_1 и x_{2i} объема n_2. При этом используется статистика $t=s_1^2/s_2^2$, где s_1^2 и s_2^2 — выборочные дисперсии случайных величин X_1 и X_2 соответственно. Для упрощения процедуры проверки гипотезы в качестве случайной величины X_1 принимаем ту случайную величину, выборочная дисперсия которой больше: $s_1^2 \geqslant s_2^2$. Отсюда имеем $t \geqslant 1$.

В зависимости от постановки задачи и выбора альтернативной гипотезы H_1 используются следующие решающие правила [20]:

а) двухсторонний критерий (гипотеза $H_1 : \sigma_1^2 \neq \sigma_2^2$). Гипотеза H_0 принимается, если $t \leqslant F_{\nu_1,\,\nu_2;\,1-\alpha/2}$, в противном случае H_0 отвергается;

б) односторонний критерий (гипотеза $H_1 : \sigma_1^2 > \sigma_2^2$). Гипотеза H_0 принимается, если $t < F_{\nu_1,\nu_2;\,1-\alpha}$, в противном случае гипотеза H_0 отклоняется;

Здесь $F_{\nu_1\nu_2 q}$ — квантиль q-го порядка F-распределения Фишера — Снедекора с (ν_1, ν_2) степенями свободы (см. табл. П.8—11), $\nu_1=n_1-1$; $\nu_2=n_2-1$.

В программах 6.29 используются предварительно вычисленные по выборкам $\{x_{1i}\}$ и $\{x_{2i}\}$, например, с помощью программ 6.2, выборочные дисперсии s_1^2 и s_2^2

Программа 6.29/21

P_2	С/П	Р3	С/П	Р4	С/П	Р5	РНОП	F2	↑	F3	÷
Р6	↑	F4	—	РХ<0	3	0	БП	F÷	1	Р7	F6
↑	F5	—	РХ<0	Р/—/	0	БП	5	1	Р8	С/П	

Программа 6.29/34

П0	С/П	П1	С/П	↑	С/П	÷	П2	↑	ИП0
—	FX<0	17	0	П3	БП	19	1	П3	ИП2
↑	ИП1	—	FX<0	27	0	С/П	1	С/П	

Инструкция к программе 6.29/21: В/О; набрать s_1^2 С/П; s_2^2 С/П; $F_{\nu_1,\nu_2;\,1-\alpha/2}$ С/П; $F_{\nu_1,\nu_2;\,1-\alpha}$ С/П; $<$Р6$>$ $=t$, $<$Р7$>$ — индикатор двухстороннего критерия, $<$Р8$>$ — одностороннего.

Инструкция к программе 6.29/34: В/О; набрать $F_{\nu_1,\,\nu_2;\,1-\alpha/2}$ С/П; $F_{\nu_1,\,\nu_2;\,1-\alpha}$ С/П; s_1^2 С/П; s_2^2 С/П; $<$Р2$>$ $=t$, $<$Р3$>$ — индикатор двухстороннего критерия, $<$РХ$>$ — одностороннего.

Проверка гипотезы о равенстве дисперсий нескольких гауссовских случайных величин по независимым выборкам. Случай выборок одинакового объема (критерий Кокрена). Пусть случайные величины X_1, ..., X_l $(l>2)$ распределены по гауссовскому закону. Извлечены l независимых выборок этих случайных величин $\{x_{1i}\}$, ..., $\{x_{2i}\}$ одинакового объема n. Требуется по этим выборкам при уровне значимости α проверить нулевую гипотезу H_0 об однородности дисперсий:

$$H_0\!:\sigma_1^2=\sigma_2^2=\ldots=\sigma_l^2.$$

Альтернативная гипотеза H_1 состоит в том, что дисперсия хотя бы одной случайной величины значимо отличается от значений дисперсий других случайных величин.

Критерий Кокрена основан на использовании статистики

$$t = \max\{s_i^2\} \Big/ \sum_{i=1}^{l} s_i^2,$$

где s_i^2 — выборочная дисперсия случайной величины X_i; $\max\{s_i^2\}$ — максимальное значение выборочной дисперсии.

Статистика t при справедливости нулевой гипотезы имеет распределение Кокрена $C_{l\nu}$ с (l, ν) степенями свободы ($\nu = n - 1$).

Гипотеза H_0 принимается, если наблюдаемое значение статистики $t < C_{l\nu\alpha}$, где $C_{l\nu\alpha}$ — квантиль порядка α распределения Кокрена $C_{l\nu}$ (см. табл. П.12).

В противном случае гипотеза H_0 отклоняется.

Если нулевая гипотеза принимается, т. е. все наблюдаемые выборки можно считать выборками из одной генеральной совокупности, то в качестве оценки дисперсии этой совокупности принимается среднее арифметическое выборочных дисперсий:

$$s^2 = \left(\sum_{i=1}^{l} s_i^2\right) / l.$$

В программах 6.30 используются предварительно вычисленные, например, с использованием программ 6.2, выборочные дисперсии s_i^2.

Программа 6.30/21

P2	С/П	P3	0	P4	P5	С/П	↑	P6	F4	+	P4
F5	—	PX ≥ 0	P3	F6	P5	С/П	БП	F1	F5	↑	F4
÷	P6	↑	F2	—	PX ≥ 0	5	1	С/П	0	P7	F4
↑	F3	÷	P8	F7	С/П						

Программа 6.30/34

0	П0	П1	П2	П4	С/П	↑	П3	ИП0	+
П0	КИП4	ИП2	ИП3	—	FX<0	05	ИП3	П2	БП
05	ИП0	ИП4	÷	П1	ИП2	ИП0	÷	П3	С/П
—	FX ≥ 0	35	1	С/П	0	С/П			

Инструкция к программе 6.30/21: В/О; набрать $C_{l\nu\alpha}$ С/П; l С/П; последовательно вводить значения s_i^2 С/П;

после ввода всех данных выполнить команды БП 3 С/П; <P4> $= \Sigma s_i^2$, <P5> $= \max\{s_i^2\}$, <P6> $= t$, <P8> $= \overline{s^2}$, <PX> — индикатор гипотезы.

Инструкция к программе 6.30/34: В/О; С/П; последовательно набирать значения s_i^2 С/П; после ввода всех данных набрать БП 21 С/П; набрать $C_{l\nu\alpha}$ С/П; <РХ> — индикатор гипотезы, <Р0> $= \Sigma s_i^2$, <Р1> $= \overline{s^2}$, <Р2> $= \max\{s_i^2\}$, <Р4> $= l$, <Р3> $= t$.

Случай выборок различного объема (критерий Бартлетта). Пусть случайные величины $X_1, \ldots, X_l$ ($l > 2$) распределены по гауссовскому закону. Извлечены l независимых выборок $\{x_{1i}\}, \ldots, \{x_{li}\}$. Объем каждой выборки равен n_i ($i = 1, \ldots, l$), причем $n_i \neq \mathrm{const}$ (если объемы всех выборок равны, то предпочтительнее использовать критерий Кокрена [20]). Требуется при уровне значимости α проверить нулевую гипотезу H_0 об однородности дисперсий:

$$H_0: \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_l^2.$$

Альтернативная гипотеза H_1 состоит в том, что хотя бы одна из случайных величин X_i ($i = 1, \ldots, l$) имеет дисперсию, значимо отличающуюся от дисперсий остальных случайных величин.

Проверка гипотезы производится с использованием статистики

$$t = (\nu \ln s^2 - b) \Big/ \left[\frac{\nu' - \nu^{-1}}{3(l-1)} + 1\right],$$

где $\nu = \Sigma\nu_i$; $\nu^1 = \Sigma 1/\nu_i$; $s^2 = (\Sigma \nu_i s_i^2)/\nu$; $b = \Sigma \nu_i \ln s_i^2)/\nu$.

Здесь $\nu_i = n_i - 1$ — число степеней свободы i-й выборки; s_i^2 — выборочная дисперсия i-й выборки.

Статистика t при справедливости нулевой гипотезы H_0 приближенно имеет χ^2-распределение с $k = l - 1$ степенями свободы, если объем каждой выборки $n_i \geqslant 4$.

Нулевая гипотеза H_0 принимается при уровне значимости α, если $t > \chi^2_{\nu;\, 1-\alpha}$, в противном случае она отклоняется. Здесь $\chi^2_{\nu q}$ — квантиль q-го порядка χ^2-распределения с ν степенями свободы (см. табл. П.6 и программы (6.26).

Если нулевая гипотеза принимается, то в качестве оценки дисперсии генеральной совокупности, из которой извлечены все l выборок, принимают среднее арифметическое выборочных дисперсий, взвешенных по числам степеней свободы выборок:

$$s^2 = \left(\sum_{i=1}^{l} \nu_i \, s_i^2 \right) / \nu .$$

Программы 6.31 позволяют по предварительно вычисленным выборочным дисперсиям s_i^2 и числу степеней свободы выборок $\nu_i = n_i - 1$ вычислить наблюдаемое значение статистики t и оценку дисперсии генеральной совокупности s^2. В программе 6.31/34 кроме того реализуется и процедура принятия решения:

Программа 6.31/21

Р7	Pln	↑	С/П	Р8	×	↑	F3	+	Р3	F8	↑
F4	+	Р4	F7	×	↑	F2	+	Р2	F8	F1/x	↑
F5	+	Р5	С/П	БП	Р0	F5	↑	F4	F1/x	—	3
÷	↑	F6	÷	1	+	Р8	F2	↑	F4	÷	Р7
Pln	↑	F4	×	↑	F3	—	↑	F8	÷	Р8	С/П

Программа 6.31/34

0	П0	П1	П2	П4	П5	С/П	П3	С/П	П6
×	ИП5	+	П5	ИП6	Fln	ИП3	×	ИП2	+
П2	ИП3	ИП0	+	П0	ИП3	F1/x	ИП1	+	П1
КИП4	БП	06	ИП5	ИП0	÷	П7	Fln	ИП0	×
ИП2	—	ИП1	ИП0	F1/x	—	3	÷	ИП4	1
—	÷	1	+	÷	П8	С/П	—	FX⩾0	62
1	С/П	0	С/П						

Инструкция к программе 6.31/21 : В/О; 0 Р2 Р3 Р4 Р5; $l - 1$ Р6; последовательно набирать s_i^2 С/П; ν_i С/П; после ввода всех данных выполнить команды БП Р5 С/П; в результате $<PX> = <P8> = t$, $<P2> = \Sigma \nu_i s_i^2$, $<P3> = b$, $<P4> = \nu$, $<P5> = \nu'$, $<P7> = s^2$.

Инструкция к программе 6.31/34: В/О С/П; последовательно набирать ν_i С/П, s_i^2 С/П; после ввода всех данных набрать БП 33 С/П, при этом $<P0> = \nu$; набрать $\chi^2_{\nu q}$ С/П; $<PX>$ — индикатор гипотезы, $<P0> = \nu$, $<P1> = \nu'$, $<P2> = b$, $<P4> = l$, $<P5> = \Sigma \nu_i s_i^2$, $<P8> = t$, $<P7> = s^2$.

Проверка гипотезы о коэффициенте линейной корреляции двухмерной гауссовской случайной величины (двух одномерных гауссовских случайных величин). Пусть двухмерная случайная величина (X, Y) имеет гауссовское распределение. Нулевая гипотеза H_0: $\rho = \rho_0$, где ρ — коэффициент корреляции компонент X и Y;

ρ_0 — гипотетическое (предполагаемое) значение корреляции. Гипотеза H_0 проверяется при уровне значимости α по результатам двухмерной выборки $\{x_i, y_i\}$ объема n с использованием статистики (z — критерий Фишера)

$$t = (z - \xi_0) \sqrt{n - 3},$$

где

$$z = \frac{1}{2} \ln \frac{1 + r_{XY}}{1 - r_{XY}}; \quad \xi_0 = \frac{1}{2} \ln \frac{1 + \rho_0}{1 - \rho_0} + \frac{\rho_0}{2(n-1)} .$$

Здесь r_{XY} — выборочный коэффициент корреляции. При справедливости нулевой гипотезы H_0 статистика t распределена асимптотически по гауссовскому закону $[N(0; 1)]$.

В зависимости от постановки задачи и вида альтернативной гипотезы H_1 различают двухсторонний и односторонние критерии:

а) двухсторонний критерий ($H_1 : \rho \neq \rho_0$). Гипотеза принимается, если $|t| < \lambda_{1-\alpha/2}$, в противном случае она отклоняется;

б) односторонний (правосторонний) критерий ($H_1 : \rho > \rho_0$). Гипотеза H_0 принимается, если $t < \lambda_{1-\alpha}$, в противном случае она отклоняется;

в) односторонний (левосторонний) критерий ($H_1 : \rho < \rho_0$). Гипотеза H_0 принимается, если $t > \lambda_{1-\alpha}$, в противном случае нулевая гипотеза отвергается.

Здесь λ_q — квантиль порядка q стандартного гауссовского распределения (см. табл. П.4).

Программа 6.32/21 позволяет по предварительно вычисленному значению r_{XY} и величинам ρ_0 и n определить наблюдаемое значение статистики t:

Программа 6.32/21

Р2	С/П	Р3	С/П	↑	1	+	Р4	2	—	/—/	F1/x
↑	F4	×	Pln	2	÷	Р4	1	↑	F2	+	Р5
2	—	/—/	F1/x	↑	F5	×	Pln	2	÷	Р5	F3
1	—	2	×	F1/x	↑	F2	×	↑	F5	+	Р5
F3	3	—	F√	Р6	F4	↑	F5	—	↑	F6	×

Программа 6.32/34

П0	С/П	П1	1	—	2	×	÷	ИП0	1
+	1	ИП0	—	÷	Fln	2	÷	+	П0
С/П	П2	1	+	1	ИП2	—	÷	Fln	2
÷	П2	XY	—	ИП1	3	—	F√	×	П1
FX<0	43	/—/	С/П	—	FX<0	51	0	П3	БП
53	1	П3	С/П	П5	ИП1	—	FX⩾0	63	0
П4	БП	65	1	П4	ИП1	ИП5	+	FX⩾0	72
0	С/П	1	С/П						

Инструкция к программе 6.32/21: В/О; набрать ρ_0 С/П; n С/П; r_{XY} С/П; <Р4> $= z$, <Р5> $= \xi_0$, <РХ> = <Р2> $= t$.

Программа 6.32/34 после ввода квантилей λ_q реализует также и процедуру принятия решения.

Инструкция к программе 6.32/34: В/О; набрать ρ_0 С/П; n С/П; r_{XY} С/П; при этом в Р0 содержится ξ_0, в Р1 — t, в Р2 — z; набрать $\lambda_{1-\alpha/2}$ С/П; $\lambda_{1-\alpha}$ С/П; в результате <Р3> — индикатор двухстороннего критерия <Р4> — правостороннего, <РХ> — левостороннего.

Проверка гипотезы о равенстве коэффициентов линейной корреляции двух двумерных гауссовских случайных величин. Пусть двумерные случайные величины (X_1, Y_1) и (X_2, Y_2) имеют двумерные гауссовские распределения. Нулевая гипотеза $H_0: \rho_1 = \rho_2$, где ρ_j ($j=1, 2$) — коэффициент корреляции j-й двумерной случайной величины. Гипотеза H_0 проверяется при уровне значимости α по результатам двух независимых двумерных выборок $\{x_{1i}, y_{1i}\}$ и $\{x_{2i}, x_{2i}\}$ объемом n_1 и n_2 соответственно с использованием статистики (z — критерий Фишера)

$$t = (z_1 - z_2)/\sqrt{\frac{1}{n_1-3} + \frac{1}{n_2-3}},$$

где

$$z_j = \frac{1}{2}\ln\frac{1+r_{XYj}}{1-r_{XYj}} \quad (j = 1, 2).$$

Здесь r_{XYj} — выборочный коэффициент корреляции, вычисленный по выборке $\{x_{ji}, y_{ji}\}$. При справедливости нулевой гипотезы статистика t распределена асимптотически по гауссовскому закону $[N(0; 1)]$.

В зависимости от альтернативной гипотезы используют следующие решающие правила:

а) двухсторонний критерий ($H_1: \rho_1 \neq \rho_2$). Нулевая гипотеза H_0 принимается, если $|t| < \lambda_{1-\alpha/2}$, в противном случае H_0 отклоняется;

б) односторонний (правосторонний) критерий ($H_1: \rho_1 > \rho_2$). Нулевая гипотеза H_0 принимается, если $t < \lambda_{1-\alpha}$, в противном случае она отвергается;

в) односторонний (левосторонний) критерий ($H_1: \rho_1 < \rho_2$). Гипотеза H_0 принимается, если $t > \lambda_{1-\alpha}$, в противном случае она отклоняется.

Здесь λ_q — квантиль порядка q стандартного гауссовского распределения (см. табл. П.4).

Программы 6.33 реализуют как расчет наблюдаемого значения статистики t по полученным предварительно выборочным коэффициентам корреляции и объемам выборок, так и процедуру принятия решения, причем в программе 6.33/34 производится проверка нулевой гипотезы при всех трех возможных видах конкурирующих гипотез H_1.

Программа 6.33/21

Р7	С/П	↑	3	—	F1/x	Р2	С/П	↑	3	—	F1/x
↑	F2	+	F√	F1/x	Р2	С/П	ПП	РСx	Р3	С/П	ПП
РСx	/—/	↑	F3	+	↑	F2	×	Р2	Fx²	F√	↑
F7	—	РХ≥0	Р7	1	С/П	0	С/П	↑	1	+	Р8
2	—	/—/	F1/x	↑	F8	×	Рln	2	÷	В/О	
									F2	↑	F7
—	РХ≥0	ВП	1	С/П	0	С/П	РНОП				
									F2	↑	F7
+	РХ≥0	ВП	0	С/П	1	С/П	РНОП				

Программа 6.33/34

0	П4	С/П	↑	↑	1	+	XY	1	XY
—	÷	Fln	2	÷	КП4	БП	02	ИП1	ИП2
—	С/П	↑	3	—	F1/x	С/П	↑	3	—
F1/x	+	F√	÷	П0	FX<0	38	/—/	С/П	—
FX<0	46	0	П3	БП	48	1	П3	С/П	П4
ИП0	—	FX≥0	58	0	П5	БП	60	1	П5
ИП0	ИП4	+	FX≥0	67	0	С/П	1	С/П	

Инструкция к программе 6.33/21: В/О; набрать $\lambda_{1-\alpha/2}$ С/П; n_1 С/П; n_2 С/П; r_{XY1} С/П; r_{XY2} С/П; <Р2> $= t$, <РХ> — индикатор двухсторонней гипотезы. Если заменить участок программы, начиная с адреса 72, на соответствующие фрагменты, то можно проверить и правосторонний и левосторонний критерий. При этом вместо $\lambda_{1-\alpha/2}$ нужно набирать значение $\lambda_{1-\alpha}$, <РХ> — индикатор критерия.

Проверку односторонних критериев можно провести и после проверки двустороннего. Для этого, не изменяя содержимого регистров памяти, ввести соответствующий фрагмент программы и набрать $\lambda_{1-\alpha}$ Р7 БП 5 С/П.

Инструкция к программе 6.33/34: В/О С/П; набрать r_{XY1} С/П; r_{XY2} С/П; БП 18 С/П; набрать n_1 С/П; n_2 С/П; $\lambda_{1-\alpha/2}$ С/П; $\lambda_{1-\alpha}$ С/П; в результате <РХ> — индикатор левостороннего критерия, <Р3> — двухстороннего, <Р5> — правостороннего, <Р1> $= z_1$, <Р2> $= z_2$, <Р0> $= t$.

Контрольный пример. Исходные данные: $n_1 = 150$, $r_{XY1} = 0{,}583$; $n_2 = 200$, $r_{XY2} = 0{,}361$; $\lambda_{1-\alpha/2} = 1{,}96$, $\lambda_{1-\alpha} =$

$=1{,}6449$ ($\alpha=0{,}05$). Результаты: наблюдаемое значение статистики $t=2{,}651255$, гипотеза H_0 принимается только при альтернативной гипотезе $H_1:\rho_1<\rho_2$, при остальных конкурирующих гипотезах нулевая гипотеза отклоняется.

6.5. Моделирование случайных величин с заданными законами распределения

Для моделирования случайных величин (случайных чисел) с требуемым законом распределения вероятностей и числовыми характеристиками, как правило, используются так называемые псевдослучайные числа. Псевдослучайные числа в основных чертах подобны соответствующим случайным величинам, однако формируются они с помощью специального детерминированного алгоритма, называемого часто генератором случайных (псевдослучайных) чисел, и в силу этого, строго говоря, случайными числами не являются. В настоящее время разработано большое число генераторов псевдослучайных чисел, имеющих различные законы распределения. Как правило, все они основаны на первоначальном получении псевдослучайных чисел, имеющих равномерное распределение в в некотором интервале. Затем эти числа преобразуются таким образом, чтобы обеспечить требуемый закон распределения.

Поскольку целочисленная арифметика изучена лучше, чем арифметика рациональных чисел, то обычно моделирование случайной величины начинают с генерации целого числа, распределенного равномерно в некоторой области неотрицательных значений. Для этого используется некоторая функция f, отображающая множество целых чисел в себя. Затем принимается начальное значение целого числа ξ_0 и каждое последующее и псевдослучайное число генерируется посредством выбранного отображения:

$$\xi_{i+1}=f(\xi_i)\quad (i=0,\ldots).$$

Известно немало различных видов отображения функций [28, 31]. В частности, широко применяется метод Коробова, в котором используется следующий алгоритм:

$$\xi_{i+1}=(q\xi_i)_{\mathrm{mod}\,p}=(q\xi_i-p\,\{q\xi_i/p\}),$$

где p — большое простое число, например, $p=2027$ или $p=5087$; q — целое число, отвечающее условиям: $q=p-3^m$ (m — целое число), $q\approx p/2$; { } — оператор выделения целой части числа.

Генерируемое таким образом число ξ_{i+1} имеет распределение, близкое к равномерному в интервале $[1, p-1]$. Далее, с помощью простой нормировки и смещения можно легко перейти к числам y_{i+1}, распределенным в произвольном интервале $[a, b]$. Этот алгоритм реализован в программах 6.34:

Программа 6.34/21

P4	F2	↑	2	÷	P7	F6	↑	F5	—	↑	F2
÷	P8	F6	↑	F5	+	↑	2	÷	P6	F4	↑
F3	×	P5	↑	F2	÷	↑	1	ВП	7	XY	+
XY	—	↑	F2	×	↑	F5	XY	—	P4	↑	F7
—	↑	F8	×	↑	F6	+	С/П	БП	÷		

Программа 6.34/34

П0	С/П	П1	2	÷	П7	С/П	П2	С/П	П3
+	2	÷	П5	ИП3	ИП2	—	ИП1	÷	П6
С/П	П4	ИП4	ИП0	×	П9	ИП1	÷	П8	1
—	FX<0	36	0	БП	38	КИП8	ИП8	ИП1	×
ИП9	XY	—	П4	ИП7	—	ИП6	×	ИП5	+
С/П	БП	22							

Инструкция к программе 6.34/21: В/О; набрать p Р2; q Р3; a Р5; b Р6; ξ_0 С/П; $\langle PX\rangle=y_i$, для расчета последующих значений y_i нажимать клавишу С/П, не изменяя содержимого регистра РХ.

Инструкция к программе 6.34/34: В/О; набрать q С/П; p С/П; a С/П; b С/П; ξ_0 С/П; в результате $\langle PX\rangle=y_i$, для расчета последующих значений нажимать клавишу С/П.

Как отмечалось, после вычисления псевдослучайного числа с равномерным законом распределения возможно такое его функциональное преобразование $y_i=\varphi(\xi_i)$, при котором обеспечивается требуемый закон распределения. Некоторые виды функциональных преобразований приведены в табл. 6.2. Здесь приняты следующие обозначения: после обозначения случайной величины с помощью буквы, характеризующей тип распределения, в скобках приведены значения параметров распределения. Например, R (0; 1) обозначает случайную величину, распределенную равномерно в интервале [0; 1]; $N(\mu, \sigma^2)$ — гауссову случайную величину с математическим ожиданием μ и дисперсией σ^2 и т. д.

Программы 6.35 позволяют получать последовательности псевдослучайных чисел $\{y_i, \xi_i, z_i\}$, распределенных

Таблица 6.2

Распределение	Преобразование
Вейбулла	$W(b,\ c) = b\,[-\log R(0,1)]^{1/c}$
Парето	$P(c) = [1/R(0,1)]^{1/c}$
Коши	$C(0,1) = \operatorname{tg}[2\pi R(0,1)]$
Эрланга	$\gamma(b,\ c) = -b\log\left[\prod\limits_{i=1}^{c} R_i(0,1)\right]$
Геометрическое	$G(p) = \log[R(0,1)]/\log(1-p)$
Рэлея	$Z(\sigma) = \sqrt{2\sigma^2 \ln[1/R(0,1)]}$
Гаусса	$N(0,\ \sigma^2) = Z(\sigma)\sin[2\pi R(0,1)]$; $N(0,\ 1) \approx \dfrac{\sum\limits_{i=1}^{k} R_i(0,\ 1) - k/2}{(k/12)^{1/2}}$
χ^2-распределение с ν степенями свободы	Для четных ν: $\chi_\nu^2 = -\dfrac{1}{2}\log\left[\prod\limits_{i=1}^{\nu/2} R_i(0,\ 1)\right]$; для нечетных ν: $\chi_\nu^2 = -\dfrac{1}{2}\log\left[\prod\limits_{i=1}^{\frac{\nu-1}{2}} R_i(0,1) + N^2(0,1)\right]$; для $\nu=1$ $\chi^2_1 = N^2(0,1)$
Логнормальное	$L(m,\ \sigma) = m\exp[\sigma^2 N(0,1)]$; $L(m,\ \sigma) \approx m\exp\left[\sigma\left(\sum\limits_{i=1}^{12} R_i(0,1) - 6\right)\right]$

соответственно по гауссовскому, равномерному и рэлеевскому законам:

Программа 6.35/21

↑	F3	×	P4	↑	F2	÷	1	ВП	7	XY	+
XY	—	↑	F2	×	↑	F4	XY	—	↑	F6	XY
P6	F2	÷	F1/x	P ln	2	×	↑	F7	×	F √	P8
F6	↑	F2	÷	↑	P π	×	2	×	P sin	↑	F8
×	↑	F5	+	С/П	F6	БП	P0				

Программа 6.35/34

П0	С/П	П1	2	÷	П7	С/П	П2	С/П	П3
С/П	П4	ИП4	ИП0	×	П5	ИП1	÷	П8	1
—	FX<0	26	0	БП	28	КИП8	ИП8	ИП1	×
ИП5	XY	—	П4	ИП7	—	ИП1	÷	2	F1/x
+	ИП6	П9	XY	П6	F1/x	F ln	ИП3	×	2
×	F √	ПА	ИП9	F π	×	F sin	ИПА	×	ИП2
+	ПВ	С/П	БП	12					

Инструкция к программе 6.35/21: В/О; набрать p Р2; q Р3; μ Р5; 1000 Р6; σ^2 Р7; $\xi_0=1$ С/П; в результате $\langle РХ\rangle = y_i$, $\langle Р6\rangle = \xi_i$, $\langle Р8\rangle = z_i$, для расчета последующих значений нажимать клавишу С/П, не изменяя содержимого регистров.

Инструкция к программе 6.35/34: В/О; набрать q С/П; p С/П; μ С/П; σ^2 С/П; $\xi_0=1$ С/П; в результате $\langle РХ\rangle = \langle РВ\rangle = y_i$; $\langle Р6\rangle = \xi_i$; [0, 1]; $\langle РА\rangle = z_i$.

Глава седьмая

СТАТИСТИЧЕСКИЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ

7.1. Временные ряды и задачи их исследования

В инженерных и экономических расчетах возникают задачи, в которых возможно наблюдение лишь одной реализации случайного процесса $y(t)$ и только в фиксированные моменты времени $t_1, \ldots, t_n$. Совокупность результатов таких наблюдений $y_1, y_2, \ldots, y_n \equiv \{y_i\}$ называется временным рядом (случайной последовательностью). В дальнейшем будем рассматривать временные ряды, наблюдаемые в равноотстоящие моменты времени, что позволяет вместо моментов времени $\{t_i\}$ использовать их номера $\{i\}$ $i=1, \ldots, n$.

Основной задачей анализа временных рядов является построение модели временного ряда и выделение компонент временного ряда. Ее решение позволяет, в частности, прогнозировать будущие и определять прошлые (недоступные непосредственному измерению) значения временных рядов.

Временные ряды могут включать в себя четыре основные компоненты:

1. Тренд или основная детерминированная тенденция временного ряда.
2. Периодическая или сезонная компонента.
3. Случайная компонента.
4. Случайная несистематическая компонента, вызванная помехами и шумами измерений.

Для выделения этих компонент и определения описывающих их параметров существует ряд методов.

Для определения тренда широко используются метод наименьших квадратов, применяемый как для выборки полного объема, т. е. для всех имеющихся в распоряжении исследователя значений временного ряда $\{y_i\}$, так и для скользящей выборки меньшего объема, описанный в § 7.2, 7.3, и метод скользящей медианы (см. § 7.4).

Для выделения периодических компонент используются методы спектрального анализа, описанные в § 5.4.

Еще одной задачей, возникающей при анализе временных рядов, является их моделирование. В связи с этим в § 7.5 приводятся алгоритмы и программы для моделирования рядов, содержащих полиномиальные тренды и случайные компоненты.

7.2. Выделение тренда временного ряда и построение его полиномиальной модели по выборке полного объема методом наименьших квадратов

Пусть наблюдаемый временной ряд $\{y_i\}$, $i=1,\dots,n$ представляется в виде суммы двух компонентов: полиномиального тренда $\{f_i = a+bi+ci^2+\dots+di^N\}$ и статистической ошибки $\{\varepsilon_i\}$:

$$y_i = f_i - \varepsilon_i \quad (i = 1, \dots, n).$$

Требуется оптимальным образом оценить параметры $a, b, c, \dots, d$ модели временного ряда и определить сглаженное значение $\hat{y}_k$, причем значение k может находиться как в интервале $1, \dots, n$ (интерполяция ряда), так и за его пределами (экстраполяция ряда).

При применении метода наименьших квадратов в качестве критерия оптимальности оценок используют условие минимума суммы квадратов невязок оценок:

$$Q = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \to \min.$$

Оценки параметров определяются решением системы уравнений

$$\begin{cases} dQ/da = 0 \\ dQ/db = 0 \\ dQ/dc = 0 \\ \cdot\ \cdot\ \cdot\ \cdot \\ dQ/dd = 0 \end{cases},$$

которая носит название системы нормальных уравнений.

Известно, что, если ошибки измерений ε_i представляют собой некоррелированные гауссовские случайные величины с одинаковыми дисперсиями и нулевыми математическими ожиданиями, то полученные оценки $\hat{a}, \hat{b}, \hat{c}, \dots, \hat{d}$ совпадают с максимально правдоподобными оценками и являются несмещенными, эффективными и состоятельными [6].

Сглаженное значение ряда $\hat{y}_k$ определяется следующим образом:

$$\hat{y}_k = \hat{a} + \hat{b}k + \hat{c}k^2 + \dots + \hat{d}k^N.$$

Часто используется линейная модель тренда

$$f_i = a + bi.$$

В этом случае система нормальных уравнений имеет вид

$$\begin{cases} na + \left(\sum\limits_{i=1}^{n} i\right) b = \sum\limits_{i=1}^{n} y_i; \\ \left(\sum\limits_{i=1}^{n} i\right) a + \left(\sum\limits_{i=1}^{n} i^2\right) b = \sum\limits_{i=1}^{n} iy_i. \end{cases}$$

Отсюда определяются выражения для оценок параметров модели ряда

$$\hat{b} = \frac{\sum\limits_{i=1}^{n} iy_i - \frac{1}{n}\sum\limits_{i=1}^{n} i \sum\limits_{i=1}^{n} y_i}{\sum\limits_{i=1}^{n} i^2 - \frac{1}{n}\left(\sum\limits_{i=1}^{n} i\right)^2};$$

$$\hat{a} = \frac{1}{n}\left(\sum_{i=1}^{n} y_i - \hat{b}\sum_{i=1}^{n} i\right).$$

Сглаженное значение ряда $\hat{y}_k$ определяется с использованием оценок параметров:

$$\hat{y}_k = \hat{a} + \hat{b}k.$$

Программы 7.1 предназначены для оценки параметров линейного тренда и для получения сглаженных значений ряда:

Программа 7.1/21

↑	F3	+	P3	XY	С/П	P7	×	↑	F4	+	P4
F7	Fx^2	↑	F5	+	P5	F7	↑	F2	+	P2	С/П
F2	Fx^2	↑	F6	÷	/—/	↑	F5	+	P5	F2	↑
F3	×	↑	F6	÷	/—/	↑	F4	+	↑	F5	÷
P5	↑	F3	×	/—/	↑	F2	+	↑	F6	÷	С/П

Программа 7.1/34

П0	П5	0	П1	П2	П3	П4	С/П	↑	↑
Fx^2	ИП4	+	П4	F→	ИП1	+	П1	XY	С/П
↑	ИП2	+	П2	F→	×	ИП3	+	П3	FL0
07	ИП5	ИП3	×	ИП1	ИП2	×	—	ИП5	ИП4
×	ИП1	Fx^2	—	÷	П6	ИП2	ИП5	÷	ИП1
ИП5	÷	ИП6	×	—	С/П				

Инструкция к программе 7.1/21: В/О; набрать 0 Р2 Р3 Р4 Р5; n Р6; поочередно набирать пары значений i С/П; y_i С/П; после ввода всех данных набрать БП Р4 С/П; значение $\hat{a}$ содержится в РХ, значение $\hat{b}$ — в Р5.

Инструкция к программе 7.1/34: В/О; набрать n С/П; поочередно набирать пары значений i С/П; y_i С/П; после ввода последней пары автоматически происходит переход к основной части программы; $\hat{a}$ содержится в РХ, $\hat{b}$ — в Р6.

Аналогично решается задача оценки параметров модели ряда и его сглаживания при квадратичной структуре тренда: $f_i = a + b_i + c_i^2$. В этом случае система нормальных уравнений имеет вид

$$\begin{cases} na + \left(\sum_{i=1}^{n} i\right) b + \left(\sum_{i=1}^{n} i^2\right) c = \sum_{i=1}^{n} y_i; \\ \left(\sum_{i=1}^{n} i\right) a + \left(\sum_{i=1}^{n} i^3\right) b + \left(\sum_{i=1}^{n} i^3\right) c = \sum_{i=1}^{n} i y_i; \\ \left(\sum_{i=1}^{n} i^2\right) a + \left(\sum_{i=1}^{n} i^2\right) b + \left(\sum_{i=1}^{n} i^4\right) c = \sum_{i=1}^{n} i^2 y_i. \end{cases}$$

Отсюда могут быть определены оценки параметров модели ряда. Соответствующие программы приведены, например, в работе [31].

7.3. Фильтрация временных рядов методом взвешенного скользящего среднего

Пусть временной ряд $\{y_i\}$ содержит две компоненты: тренд $\{f_i\}$ и статистическую ошибку (ошибку измерения) $\{\xi_i\}$:

$$y_i = f_i + \xi_i \quad (i = 1, \ldots, N).$$

Здесь $f_i = f(i)$ — некоторая функция номера отсчета i, соответствующего моменту проведения измерения t_i. В частности, широкий класс процессов может быть описан с использованием полиномиального представления, которое перепишем в виде

$$f(i) = a_N i^N + a_{N-1} i^{N-1} + \ldots + a_0,$$

где N — степень полинома.

Задача фильтрации состоит в том, чтобы получить последовательность оценок $\{x_i\}$, которая была бы ближе в определенном смысле к последовательности $\{f_i\}$ $(i = 1, \ldots, n)$.

Будем считать, что ошибки измерений ξ_i являются некоррелированными гауссовскими случайными величинами с нулевыми математическими ожиданиями и равными дисперсиями. Как и в § 7.2, в качестве критерия оптимальности оценок $\{x_i\}$ будем использовать сумму квадратов невязок фильтрации:

$$Q = \sum_{i=1}^{n} (x_i - y_i)^2 \to \min. \qquad (7.1)$$

В статистике, информатике, экономике и в других областях широко используется метод скользящего взвешенного среднего [12, 21]. В цифровой фильтрации ему соответствует нерекурсивный фильтр с импульсной характеристикой конечной длительности, равной m отсчетам [12]. Такой фильтр формирует оценки вида

$$x_i = \sum_{j=1}^{m} b_j y_{i+p-m+j}, \qquad (7.2)$$

где m — размер скользящего окна (длина импульсной характеристики); b_j — некоторые коэффициенты, которые выбираются исходя из условия минимизации критерия, определяемого выражением (7.1).

При $p=0$ решается задача собственно фильтрации — получения оценки текущего значения тренда по текущему и $(m-1)$ предыдущим значениям временного ряда. При $p>0$ производится сглаживание или статистическая интерполяция — формирование оценки тренда в прошедший момент времени. При $p<0$ решается задача статистической экстраполяции или прогнозирования тренда временного ряда на $|p|$ отсчетов вперед.

Как правило, длина импульсной характеристики фильтра m выбирается нечетной, причем должно выполняться условие: $m>n+1$. Значения коэффициентов $\{b_i\}$ для некоторых значений n, m и p приведены в табл. 7.1—

Таблица 7.1

Коэффициенты нерекурсивного цифрового фильтра при интерполяции на середину скользящей выборки: $N=2—3$, $p=(m—1)/2$

b_j	Длина импульсной характеристики						
	5	7	9	11	13	15	17
b_1	—3/35	—2/21	—21/231	—33/429	—11/143	—78/1105	21/323
b_2	12/35	3/21	14/231	9/429	0/143	—13/1105	—6/323
b_3	17/35	6/21	39/231	44/429	9/143	42/1105	7/323
b_4	—	7/21	54/231	69/429	16/143	87/1105	18/323
b_5	—	—	59/231	84/429	21/143	122/1105	27/323
b_6	—	—	—	89/429	24/143	147/1105	34/323
b_7	—	—	—	—	25/143	162/1105	39/323
b_8	—	—	—	—	—	167/1105	42/323
b_9	—	—	—	—	—	—	43/323

Примечание: $b_j=b_{m-j+1}$, $j=1, \ldots, (m+1)/2$.

7.3. Более полные таблицы значений коэффициентов содержатся в работах [12, 21].

Известно, что при гауссовой статистике ошибок (наблюдений) ξ_i оценки вида (7.2) совпадают с оценками

Таблица 7.2

Коэффициенты нерекурсивиого цифрового фильтра при оценке текущего значения временного ряда $N=3$; $p=0$

b_j	Длина имульсной характеристики						
	5	7	9	11	13	15	17
b_1	—1/70	—2/42	—7/99	—12/143	—33/364	—286/3060	—91/969
b_2	4/70	4/42	8/99	8/143	12/364	44/3060	0/969
b_3	—6/70	1/42	8/99	13/143	30/364	209/3060	52/969
b_4	4/70	—4/42	0/99	8/143	28/364	244/3060	72/969
b_5	69/70	—4/42	—9/99	—2/143	13/364	184/3060	67/969
b_6	—	8/42	—12/99	—12/143	—8/364	64/3060	44/969
b_7	—	39/42	—2/99	—17/143	—28/364	—81/3060	10/969
b_8	—	—	28/99	—12/143	—40/364	—216/3060	—28/969
b_9	—	—	85/99	8/143	—37/364	—308/3060	—63/969
b_{10}	—	—	—	48/143	—12/364	—316/3060	88/969
b_{11}	—	—	—	113/143	42/364	—211/3060	—96/969
b_{12}	—	—	—	—	132/364	44/3060	—80/969
b_{13}	—	—	—	—	265/364	484/3060	—33/969
b_{14}	—	—	—	—	—	1144/3060	52/969
b_{15}	—	—	—	—	—	2059/3060	182/969
b_{16}	—	—	—	—	—	—	364/969
b_{17}	—	—	—	—	—	—	605/969

Таблица 7.3

Коэффициенты нерекурсивного цифрового фильтра при экстраполяции на один шаг: $N=3$; $p=—1$

b_j	Длина импульсной характеристики						
	5	7	9	11	13	15	17
b_1	—4/5	—4/7	—56/126	—24/66	—44/143	—364/1365	—16/68
b_2	11/5	6/7	49/126	12/66	11/143	26/1365	—1/68
b_3	—4/5	4/7	64/126	24/66	36/143	236/1365	8/68
b_4	—14/5	—3/7	24/126	19/66	38/143	301/1365	12/68
b_5	16/5	—8/7	—36/126	4/66	24/143	256/1365	12/68
b_6	—	—4/7	—81/126	—14/66	1/143	136/1365	9/68
b_7	—	16/7	—76/126	—28/66	—24/143	—24/1365	4/68
b_8	—	—	14/126	—31/66	—44/143	—189/1365	—2/68
b_9	—	—	224/126	—16/66	—52/143	—324/1365	—8/68
b_{10}	—	—	—	24/66	—41/143	—394/1365	—13/68
b_{11}	—	—	—	96/66	—4/143	—364/1365	—16/68
b_{12}	—	—	—	—	66/143	—199/1365	—16/68
b_{13}	—	—	—	—	176/143	136/1365	—12/68
b_{14}	—	—	—	—	—	676/1365	—3/68
b_{15}	—	—	—	—	—	1456/1365	12/68
b_{16}	—	—	—	—	—	—	34/68
b_{17}	—	—	—	—	—	—	64/68

по методу максимального правдоподобия и являются несмещенными эффективными и состоятельными [8]. Дисперсию оценки (дисперсию ошибки) фильтрации σ_x^2 можно определить по формуле

$$\sigma_x^2 \cong \sigma^2 \sum_{j=1}^{m} b_j^2,$$

где σ^2 — дисперсия шума ξ_i.

Программы 7.2 предназначены для статистической интерполяции тренда — полинома третьей степени с отставанием на два шага при $m=5$:

Программа 7.2/21

```
P2   С/П  P3   С/П  P4   С/П  P5   С/П  P6   ↑    F2   +
↑    3    ×    P→   F3   ↑    F5   +    ↑    1    2    ×
P→   F4   ↑    1    7    ×    ↑    P←   +    ↑    P←   —
↑    3    5    ÷    P→   F3   P2   F4   P3   F5   P4   F6
P5   P←   С/П  БП   PXY
```

Программа 7.2/34

П0	С/П	П1	С/П	П2	С/П	П3	С/П	П4	ИП1
ИП3	+	1	2	×	XY	ИП0	+	3	×
—	ИП2	1	7	×	+	3	5	÷	П5
ИП1	П0	ИП2	П1	ИП3	П2	ИП4	П3	ИП5	БП
07									

Инструкция к программам 7.2: В/О; поочередно набирать значения y_i С/П; после ввода пятого и последующих отсчетов в РХ содержится значение x_{i-2}.

Программы 7.3 реализуют алгоритм фильтрации полинома той же степени на текущий момент времени при $m=5$:

Программа 7.3/21

P2	С/П	P3	С/П	P4	С/П	P5	С/П	P6	↑	6	9
×	P7	F5	4	×	↑	F7	+	P7	F4	6	/—/
×	↑	F7	+	P7	F3	4	×	↑	F7	+	↑
F2	—	↑	7	0	÷	P7	F3	P2	F4	P3	F5
P4	F6	P5	F7	С/П	БП	PXY					

Программа 7.3/34

П0	С/П	П1	С/П	П2	С/П	П3	С/П	П4	↑
6	9	×	ИП3	4	×	+	ИП2	6	×
—	ИП1	4	×	+	ИП0	—	7	0	÷
ИП1	П0	ИП2	П1	ИП3	П2	ИП4	П3	БП	07

Инструкция к программам 7.3: В/О; поочередно набирать значения y_i С/П; после ввода пятого и последующих отсчетов в РХ содержится значение x_i.

Программы 7.4 используются для статистической экстраполяции полинома той же степени на один отсчет вперед при $m=5$:

Программа 7.4/21

P2	С/П	P3	С/П	P4	С/П	P5	С/П	P6	↑	1	6
×	P→	F5	↑	1	4	/—/	×	↑	P←	+	P→
F4	↑	F2	+	↑	4	/—/	×	↑	P←	+	P→
F3	↑	1	1	×	↑	P←	+	↑	5	÷	P→
F3	P2	F4	P3	F5	P4	F6	P5	P←	С/П	БП	PXY

Программа 7.4/34

П0	С/П	П1	С/П	П2	С/П	П3	С/П	П4	1
6	×	ИП3	1	4	×	—	ИП0	ИП2	+
4	×	—	ИП1	1	1	×	+	5	÷
П5	ИП1	П0	ИП2	П1	ИП3	П2	ИП4	П3	ИП5
БП	07								

Инструкция к программам 7.4: В/О; поочередно набирать значения y_i С/П; после ввода пятого и последующих отсчетов в РХ содержится значение x_{i+1}.

Программы 7.4 используются для статистической интерполяции полинома третьей степени с отставанием на три отсчета при $m=7$:

Программа 7.5/21

F2	↑	F8	+	2	×	P→	F3	↑	F7	+	3
×	P→	F4	↑	F6	+	6	×	P→	F5	7	×
↑	P←	+	↑	P←	+	↑	P←	—	2	1	÷
P→	F3	P2	F4	P3	F5	P4	F6	P5	F7	P6	F8
P7	P←	С/П	P8	БП	P0						

Программа 7.5/34

П0	С/П	П1	С/П	П2	С/П	П3	С/П	П4	С/П
П5	С/П	П6	ИП0	+	2	×	ИП1	ИП5	+
3	×	XY	—	ИП2	ИП4	+	6	×	+
ИП3	7	×	+	2	1	÷	П7	ИП1	П0
ИП2	П1	ИП3	П2	ИП4	П3	ИП5	П4	ИП6	П5
ИП7	БП	11							

Инструкция к программе 7.5/21: В/О; набрать y_1 Р2; y_2 Р3; y_3 Р4; y_4 Р5; y_5 Р6; y_6 Р7; y_7 Р8; нажать клавишу С/П, при этом в РХ содержится значение x_4; в дальнейшем поочередно вводятся значения y_i С/П; в РХ содержатся оценки x_{i-3}.

Инструкция к программе 7.5/34: В/О; поочередно набирать y_i С/П; после ввода седьмого и последующих значений в РХ содержится оценка x_{i-3}.

7.4. Фильтрация временных рядов методом скользящей медианы

Одним из недостатков метода взвешенного скользящего среднего является сравнительно низкая устойчивость оценок к так называемым аномальным ошибкам измерений элементов временного ряда. Аномальные ошибки могут происходить по многим причинам — из-за грубых ошибок (промахов) наблюдателей, из-за воздействия импульсных помех и т. д. При этом закон распределения ошибок перестает быть гауссовским. В таких случаях необходимо использовать так называемые робастные или устойчивые методы фильтрации и оценивания [21]. Одним из наиболее распространенных методов робастной фильтрации является фильтрация по методу скользящей медианы. Она заключается в том, что в качестве оценки x_i элемента временного ряда f_i используется выборочная медиана — центральный элемент последовательности из m отсчетов y_i, расположенных в порядке возрастания (в виде вариационного ряда):

$$x_i = \mathrm{Med}\,\{y_{i-p}, y_{i-p+1}, \ldots, y_{i-p+m}\}, \tag{7.3}$$

где p — размер скользящего окна.

Как правило, размер скользящей выборки принимают небольшим ($m=3$—7) и решают задачу интерполяции в точке, соответствующей середине скользящего окна: $p=(m-1)/2$ [22].

Оценки вида (7.3) совпадают с оценками по методу максимального правдоподобия в случае, когда шум измерений имеет распределение Лапласа, обладающее более тяжелыми «хвостами», чем гауссовское. Однако эти оценки используют часто и в других случаях, когда статистика шума отличается от гауссовой [21].

Программы 7.6 предназначены для медианной фильтрации временного ряда при $m=3$, $p=1$:

Программа 7.6/21

С/П	Р2	↑	F3	—	РХ≥0	F,	F2	Р7	↑	F4	—
РХ≥0	Сx	F3	↑	F4	—	РХ≥0	Р4	F3	Р7	БП	Сx
F4	Р7	БП	Сx	F3	Р7	↑	F4	—	РХ≥0	Сx	F4
↑	F2	—	РХ≥0	Сx	F4	Р7	БП	Сx	F2	Р7	F3
Р4	F2	Р3	F7	БП	Р0						

Программа 7.6/34

0	П0	П1	С/П	П2	ИП0	—	FX≥0	24	ИП2
П3	ИП1	—	FX≥0	41	ИП0	ИП1	—	FX≥0	38
ИП0	П3	БП	41	ИП0	П3	ИП1	—	FX≥0	41
ИП1	ИП2	—	FX≥0	41	ИП1	П3	БП	41	ИП1
П3	ИП0	П1	ИП2	П0	ИП3	БП	03		

Инструкция к программам 7.6: В/О; набрать 0 Р3 Р4 (для Б3-34 требуется только нажать клавишу С/П); поочередно набирать значения y_i С/П (в Б3-21 после набора первого значения команду С/П выполнить дважды), после ввода третьего и последующих значений в РХ содержится значение x_i (в Б3-21 это значение содержится также и в Р7).

Для обработки данных с аномальными ошибками измерений целесообразно использовать комбинированные алгоритмы обработки. В частности, сначала для удаления аномальных измерений данные могут обрабатываться с помощью медианного фильтра (программы 7.6), а затем — линейным нерекурсивным фильтром (программы 7.2—7.5).

7.5. Моделирование временных рядов

Моделирование временных рядов широко используется, например, при оценке эффективности алгоритмов обработки временных рядов.

Распространенной моделью детерминированной компоненты (тренда) временного ряда является полиномиальное представление

$$f(t_i) = a_0' + a_1' t_i + a_2' t_i^2 + \ldots + a_N' t_i^N \quad (i=1, \ldots, n),$$

где a_j' — некоторые коэффициенты.

Если временные отсчеты имеют постоянный шаг, то часто переходят к эквивалентному полиному

$$f_i = a_0 + a_i i + a_1 i^2 + \ldots + a_N i^N \quad (i=1, \ldots, n).$$

Непосредственное вычисление одного значения полиномов $f(t_i)$ или f_i требует $n(n+1)/2$ умножений и n сложений. Для сокращения числа операций, что важно при решении задач на микрокалькуляторах, целесообразно использовать прием, называемый схемой Горнера и состоящий в преобразовании исходных выражений к виду [34, 41]

$$f(t_i) = a_0' + t_i\left(a_1' + t_i\left(a_2' + t_i\left(\ldots\left(a_{N-1}' + a_N' t_i\right)\ldots\right)\right)\right).$$

Схема Горнера требует выполнения только n умножений и n сложений.

Программы 7.7 позволяют генерировать последовательность значений полинома третьей степени в равноудаленных точках, взятых с шагом Δt:

Программа 7.7/21

Р6	0	Р8	F6	↑	F2	×	↑	F3	+	↑	F6
×	↑	F4	+	↑	F6	×	↑	F5	+	С/П	F6
↑	F7	+	Р6	F8	1	+	Р8	F6	БП	F↑	

Программа 7.7/34

П0	0	П6	С/П	П1	С/П	П2	С/П	П3	С/П
П4	С/П	П5	ИП1	×	ИП5	Fx^2	ИП2	×	+
ИП5	↑	Fx^2	×	ИП3	×	+	ИП0	+	С/П
КИП6	ИП5	ИП4	+	БП	12				

Инструкция к программе 7.7/21: В/О; набрать a_3' Р2; a_2' Р3; a_1' Р4; a_0' Р5; Δt Р7; набрать t_1 С/П; последовательно нажимать клавишу С/П, при этом в РХ содержится f_i, в Р6 — t_i, в Р8 — i.

Инструкция к программе 7.7/34: В/О; набрать a_0' С/П; a_1' С/П; a_2' С/П; a_3 С/П; Δt С/П; последовательно нажимать клавишу С/П, при этом в РХ содержится f_i, в Р5 — t_i, в Р6 — i.

Для моделирования случайной компоненты временного ряда с требуемым законом распределения и заданными числовыми характеристиками могут использоваться программы, приведенные в § 6.5. Они позволяют генерировать статистически независимые отсчеты. Если необходимо моделировать коррелированный случайный процесс, то можно сначала получить реализацию некоррелированного шума, а затем профильтровать его с помощью одного из рассмотренных ранее сглаживающих фильтров. Процесс на выходе фильтра будет иметь корреляционную функцию, определяемую импульсной характеристикой фильтра.

Приложение

Таблица П.1

Плотность вероятности стандартного гауссовского распределения $\varphi(x)=\dfrac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$

x	$\varphi(x)$	x	$\varphi(x)$	x	$\varphi(x)$	x	$\varphi(x)$
0,0	0,3989	1,0	0,2420	2,0	0,0540	3,0	0,0044
0,1	0,3970	1,1	0,2170	2,1	0,0440	3,1	0,0033
0,2	0,3910	1,2	0,1942	2,2	0,0355	3,2	0,0024
0,3	0,3814	1,3	0,1714	2,3	0,0283	3,3	0,0017
0,4	0,3683	1,4	0,1497	2,4	0,0224	3,4	0,0012
0,5	0,3521	1,5	0,1295	2,5	0,0175	3,5	0,0009
0,6	0,3332	1,6	0,1109	2,6	0,0136	3,6	0,0006
0,7	0,3123	1,7	0,0940	2,7	0,0104	3,7	0,0004
0,8	0,2897	1,8	0,0790	2,8	0,0079	3,8	0,0003
0,9	0,2661	1,9	0,0644	2,9	0,0060	3,9	0,0002

Примечание: В таблице даны значения плотности вероятности только для $x\geqslant 0$, поскольку для стандартного гауссовского распределения $\varphi(x)=\varphi(-x)$.

Таблица П.2

Функция стандартного гауссовского распределения

$$\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}\,dt$$

x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$
—4,0	0,00003	—3,0	0,00135	—2,0	0,02275	—1,0	0,15866
—3,9	0,00005	—2,9	0,00187	—1,9	0,02872	—0,9	0,18406
—3,8	0,00007	—2,8	0,00256	—1,8	0,03593	—0,8	0,21186
—3,7	0,00011	—2,7	0,00347	—1,7	0,04457	—0,7	0,24196
—3,6	0,00016	—2,6	0,00466	—1,6	0,05480	—0,6	0,27425
—3,5	0,00023	—2,5	0,00621	—1,5	0,06681	—0,5	0,30854
—3,4	0,00034	—2,4	0,00820	—1,4	0,08076	—0,4	0,34458
—3,3	0,00048	—2,3	0,01072	—1,3	0,09680	—0,3	0,38209
—3,2	0,00069	—2,2	0,01390	—1,2	0,11507	—0,2	0,42074
—3,1	0,00097	—2,1	0,01786	—1,1	0,13567	—0,1	0,46017

x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$
0,0	0,50000	1,0	0,84134	2,0	0,97725	3,0	0,99865
0,1	0,53983	1,1	0,86433	2,1	0,98214	3,1	0,99903
0,2	0,57926	1,2	0,88493	2,2	0,98610	3,2	0,99931
0,3	0,61791	1,3	0,90320	2,3	0,98928	3,3	0,99952
0,4	0,65542	1,4	0,91924	2,4	0,99180	3,4	0,99966
0,5	0,69146	1,5	0,93319	2,5	0,99379	3,5	0,99977
0,6	0,72575	1,6	0,94520	2,6	0,99534	3,6	0,99984
0,7	0,75804	1,7	0,95543	2,7	0,99653	3,7	0,99989
0,8	0,78814	1,8	0,96407	2,8	0,99744	3,8	0,99993
0,9	0,81594	1,9	0,97128	2,9	0,99813	3,9	0,99995

Таблица П.3

Критическая точка Z_{nq} для критерия знаков

Объем выборки n	q, α $(q=\alpha)$ / α $(q=\alpha/2)$					Объем выборки n	q, α $(q=\alpha)$ / α $(q=\alpha/2)$				
	0,005 / 0,01	0,01 / 0,02	0,02 / 0,04	0,025 / 0,05	0,05 / 0,1		0,005 / 0,01	0,01 / 0,02	0,02 / 0,04	0,025 / 0,05	0,05 / 0,1
5	—	—	—	—	0	31	7	8	9	9	10
6	—	—	0	0	0	32	8	8	9	9	10
7	—	0	0	0	0	33	8	9	10	10	11
8	0	0	0	0	1	34	9	9	10	10	11
9	0	0	1	1	1	35	9	10	10	11	12
10	0	0	1	1	1	36	9	10	11	11	12
11	0	1	1	1	2	37	10	10	11	12	13
12	1	1	2	2	2	38	10	11	12	12	13
13	1	1	2	2	2	39	11	11	12	12	13
14	1	2	2	2	3	40	11	12	13	13	14
15	2	2	3	3	3	41	11	12	13	13	14
16	2	2	3	3	4	42	12	13	13	14	15
17	2	3	3	4	4	43	12	13	14	14	15
18	3	3	4	4	5	44	13	13	14	15	16
19	3	4	4	4	5	45	13	14	15	15	16
20	3	4	4	5	5	46	13	14	15	15	16
21	4	4	5	5	6	47	14	15	16	16	17
22	4	5	5	5	6	48	14	15	16	16	17
23	4	5	6	6	7	49	15	15	16	17	18
24	5	5	6	6	7	50	15	16	17	18	19
25	5	6	6	7	7	51	15	16	17	18	19
26	6	6	7	7	8	52	16	17	18	18	19
27	6	7	7	7	8	53	16	17	18	18	20
28	6	7	8	8	9	54	17	18	18	19	20
29	7	7	8	8	9	55	17	18	19	19	20
30	7	8	8	9	10	56	17	18	19	20	21

Объем выборки n	q, α $(q=\alpha)$ / α $(q=\alpha/2)$					Объем выборки n	q, α $(q=\alpha)$ / α $(q=\alpha/2)$				
	0,005 / 0,01	0,01 / 0,02	0,02 / 0,04	0,025 / 0,05	0,05 / 0,1		0,005 / 0,01	0,01 / 0,02	0,02 / 0,04	0,025 / 0,05	0,05 / 0,1
57	18	19	20	20	21	79	27	28	29	30	31
58	18	19	20	21	22	80	28	29	30	30	32
59	19	20	21	21	22	81	28	30	31	31	33
60	19	20	21	21	23	82	28	30	31	31	33
61	20	20	22	22	23	83	29	30	31	32	33
62	20	21	22	22	24	84	29	30	32	32	33
63	20	21	22	23	24	85	30	31	32	32	34
64	21	22	23	23	24	86	30	31	33	33	34
65	21	22	23	24	25	87	31	32	33	33	35
66	22	23	24	24	25	88	31	32	33	34	35
67	22	23	24	25	26	89	31	33	34	34	36
68	22	23	25	25	26	90	32	33	34	35	36
69	23	24	25	25	27	91	32	33	35	35	37
70	23	24	25	26	27	92	33	34	35	36	37
71	24	25	26	26	28	93	33	34	36	36	38
72	24	25	26	27	28	94	34	35	36	37	38
73	25	26	27	27	28	95	34	35	37	37	38
74	25	26	27	28	29	96	34	36	37	37	39
75	25	26	28	28	29	97	35	36	37	38	39
76	26	27	28	28	30	98	36	37	38	39	40
77	26	27	29	29	30	99	36	37	38	39	40
78	27	28	29	29	31	100	36	37	39	39	41

Таблица П.4

Квантиль λ_q гауссовского распределения

q, ε $(q=\varepsilon)$	0,9	0,95	0,975	0,99	0,995	0,999	0,9995
α $(q=1-\alpha)$	0,1	0,05	0,025	0,01	0,005	0,001	0,0005
α $(q=1-\alpha/2)$	0,2	0,1	0,05	0,02	0,01	0,002	0,001
ε $(q=1+\varepsilon/2)$	0,8	0,9	0,95	0,98	0,99	0,998	0,999
λ_q	1,2816	1,6449	1,9600	2,3263	2,5758	3,0902	3,2905

Таблица П.5

Квантиль $t_{\nu q}$ распределения Стьюдента (t-распределения)

Число степеней свободы ν	q, ε $(q=\varepsilon)$ / ε $[q=(1+\varepsilon)/2]$: 0,9 / 0,8	0,95 / 0,9	0,975 / 0,95	0,99 / 0,98	0,995 / 0,99	0,999 / 0,998	0,995 / 0,999
	α $(q=1-\alpha)$ / α $(q=1-\alpha/2)$: 0,1 / 0,2	0,05 / 0,1	0,025 / 0,05	0,01 / 0,02	0,005 / 0,01	0,001 / 0,002	0,0005 / 0,001
1	3,078	6,314	12,706	31,821	63,657	318,309	636,619
2	1,886	2,920	4,303	6,965	9,925	22,327	31,599
3	1,638	2,353	3,182	4,541	5,841	10,215	12,924
4	1,533	2,132	2,776	3,747	4,604	7,173	8,610
5	1,476	2,015	2,571	3,365	4,032	5,893	6,869
6	1,440	1,943	2,447	3,143	3,707	5,208	5,959
7	1,415	1,895	2,365	2,998	3,499	4,785	5,408
8	1,397	1,860	2,306	2,896	3,355	4,501	5,041
9	1,383	1,833	2,262	2,821	3,250	4,501	4,781
10	1,372	1,812	2,228	2,764	3,169	4,144	4,587
11	1,363	1,796	2,201	2,718	3,106	4,025	4,437
12	1,356	1,782	2,179	2,681	3,055	3,930	4,318
13	1,350	1,771	2,160	2,650	3,012	3,852	4,221
14	1,345	1,761	2,145	2,624	2,977	3,787	4,140
15	1,341	1,753	2,131	2,602	2,947	3,733	4,073
16	1,337	1,746	2,120	2,583	2,921	3,686	4,015
17	1,333	1,740	2,110	2,567	2,898	3,646	3,965
18	1,330	1,734	2,101	2,552	2,878	3,610	3,922
19	1,328	1,729	2,093	2,539	2,861	3,579	3,883
20	1,325	1,725	2,086	2,528	2,845	3,552	3,850
21	1,323	1,721	2,080	2,518	2,831	3,527	3,819
22	1,321	1,717	2,074	2,508	2,819	3,505	3,792
23	1,319	1,714	2,069	2,500	2,807	3,485	3,768
24	1,318	1,711	2,064	2,492	2,797	3,467	3,745
25	1,316	1,708	2,060	2,485	2,787	3,450	3,725
26	1,315	1,706	2,056	2,479	2,779	3,435	3,707
27	1,314	1,703	2,052	2,473	2,771	3,421	3,690
28	1,313	1,701	2,048	2,467	2,763	3,408	3,674
29	1,311	1,699	2,045	2,462	2,756	3,396	3,659
30	1,310	1,697	2,042	2,457	2,750	3,385	3,646
32	1,309	1,694	2,037	2,449	2,738	3,365	3,622
34	1,307	1,691	2,032	2,441	2,728	3,348	3,601
36	1,306	1,688	2,028	2,434	2,719	3,333	3,582
38	1,304	1,686	2,024	2,429	2,712	3,319	3,566
40	1,303	1,684	2,021	2,423	2,704	3,307	3,551
45	1,301	1,679	2,014	2,412	2,690	3,281	3,520
50	1,299	1,676	2,009	2,403	2,678	3,261	3,496

Продолжение табл. П.5

Число степеней свободы ν	q, ε $(q=\varepsilon)$ / ε $[q=(1+\varepsilon)/2]$: 0,9 / 0,8	0,95 / 0,9	0,975 / 0,95	0,99 / 0,98	0,995 / 0,99	0,999 / 0,998	0,995 / 0,999
	α $(q=1-\alpha)$ / α $(q=1-\alpha/2)$: 0,1 / 0,2	0,05 / 0,1	0,025 / 0,05	0,01 / 0,02	0,005 / 0,01	0,001 / 0,002	0,0005 / 0,001
55	1,297	1,673	2,004	2,396	2,668	3,245	3,476
60	1,296	1,671	2,000	2,390	2,660	3,232	3,460
80	1,292	1,664	1,990	2,374	2,639	3,195	3,416
90	1,291	1,662	1,987	2,368	2,632	3,183	3,402
100	1,290	1,660	1,984	2,364	2,626	3,174	3,390
120	1,289	1,658	1,980	2,358	2,617	3,160	3,373
150	1,287	1,655	1,976	2,351	2,609	3,145	3,357
200	1,286	1,653	1,972	2,345	2,601	3,131	3,340
500	1,283	1,648	1,965	2,334	2,586	3,107	3,310
	1,282	1,645	1,960	2,326	2,576	3,090	3,291

Таблица П.6

Квантиль $\chi^2_{\nu q}$ χ^2-распределения

Число степеней свободы ν	q, $\alpha(q=\alpha)$ / α $(q=\alpha/2)$: 0,001 / 0,002	0,005 / 0,01	0,01 / 0,02	0,025 / 0,05	0,05 / 0,1	0,1 / 0,2
	ε $(q=1-\varepsilon)$ / ε $[q=(1-\varepsilon)/2]$: 0,999 / 0,998	0,995 / 0,99	0,99 / 0,98	0,975 / 0,95	0,95 / 0,9	0,9 / 0,8
1	$0{,}157\times10^{-5}$	$0{,}393\times10^{-4}$	$0{,}157\times10^{-}$	$0{,}982\times10^{-3}$	$0{,}393\times10^{-2}$	0,0158
2	$0{,}200\times10^{-2}$	0,0100	0,0201	0,0506	0,103	0,211
3	0,0243	0,0717	0,115	0,216	0,352	0,584
4	0,0908	0,207	0,297	0,484	0,711	1,06
5	0,210	0,412	0,554	0,831	1,15	1,61
6	0,381	0,676	0,872	1,24	1,64	2,20
7	0,598	0,989	1,24	1,69	2,17	2,83
8	0,857	1,34	1,64	2,18	2,73	3,49

Число степеней свободы ν	q, α (q=α) / α (q=α/2): 0,001 / 0,002; ε (q=1−ε) / ε [q=(1−ε)/2]: 0,999 / 0,988	0,005 / 0,01; 0,995 / 0,99	0,01 / 0,02; 0,99 / 0,98	0,025 / 0,05; 0,975 / 0,95	0,05 / 0,1; 0,95 / 0,9	0,1 / 0,2; 0,9 / 0,8
9	1,15	1,73	2,09	2,70	3,33	4,17
10	1,48	2,16	2,56	3,25	3,94	4,87
11	1,83	2,60	3,05	3,82	4,57	5,58
12	2,21	3,07	3,57	4,40	5,23	6,30
13	2,62	3,57	4,11	5,01	5,89	7,04
14	3,04	4,07	4,66	5,63	6,57	7,79
15	3,48	4,60	5,23	6,26	7,26	8,55
16	3,94	5,14	5,81	6,91	7,96	9,31
17	4,42	5,70	6,41	7,56	8,67	10,09
18	4,90	6,26	7,01	8,23	9,39	10,86
19	5,41	6,84	7,63	8,91	10,12	11,65
20	5,92	7,43	8,26	9,59	10,85	12,44
21	6,45	8,03	8,90	10,28	11,59	13,24
22	6,98	8,64	9,54	10,98	12,34	14,04
23	7,53	9,26	10,20	11,69	13,09	14,85
24	8,08	9,89	10,86	12,40	13,85	15,66
25	8,65	10,52	11,52	13,12	14,61	16,47
26	9,22	11,16	12,20	13,84	15,38	17,29
27	9,80	11,81	12,88	14,57	16,15	18,11
28	10,39	12,46	13,56	15,31	16,93	18,94
29	10,99	13,12	14,26	16,05	17,71	19,77
30	11,59	13,79	14,95	16,79	18,49	20,60
32	12,81	15,14	16,36	18,29	20,07	22,27
34	14,06	16,50	17,79	19,81	21,66	23,95
36	15,32	17,89	19,23	21,34	23,27	25,64
38	16,61	19,29	20,69	22,88	24,88	27,34
40	17,92	20,71	22,16	24,43	26,51	29,05
42	19,24	22,14	23,65	26,00	28,14	30,77
44	20,58	23,58	25,15	27,57	29,79	32,49
46	21,93	25,04	26,66	29,16	31,44	34,22
48	23,29	26,51	28,18	30,75	33,10	35,95
50	24,67	27,99	29,71	32,36	34,76	37,69
55	28,17	31,73	33,57	36,40	38,96	42,06
60	31,74	35,53	37,48	40,48	43,19	46,46
65	37,36	39,38	41,44	44,60	47,45	50,88
70	39,04	43,28	45,44	48,76	51,74	55,33

Число степеней свободы ν	q, α (q=α) / α (q=α/2): 0,001 / 0,002; ε (q=1−ε) / ε [q=(1−ε)/2]: 0,999 / 0,988	0,005 / 0,01; 0,995 / 0,99	0,01 / 0,02; 0,99 / 0,98	0,025 / 0,05; 0,975 / 0,95	0,05 / 0,1; 0,95 / 0,9	0,1 / 0,2; 0,9 / 0,8
75	42,76	47,21	49,48	52,94	56,05	59,79
80	46,52	51,17	53,54	57,15	60,39	64,28
85	50,32	55,17	57,63	61,39	64,75	68,78
90	54,16	59,20	61,75	65,65	69,13	73,29
95	58,02	63,25	65,90	69,92	73,52	77,82
100	61,92	67,33	70,06	74,22	77,93	82,36

Таблица П.7

Квантиль $\chi^2_{\nu q}$ χ-распределения

Число свободных степеней ν	q, ε (q=ε) / ε [q=(1+ε)/2]: 0,9 / 0,8; α (q=1−α/2) / α (q=1−α): 0,2 / 0,1	0,95 / 0,9; 0,1 / 0,05	0,975 / 0,95; 0,05 / 0,025	0,99 / 0,98; 0,02 / 0,01	0,995 / 0,99; 0,01 / 0,005	0,999 / 0,998; 0,02 / 0,001
1	2,71	3,84	5,02	6,63	7,88	10,83
2	4,61	5,99	7,38	9,21	10,60	13,82
3	6,25	7,81	9,35	11,34	12,84	16,27
4	7,78	9,49	11,14	13,28	14,86	18,47
5	9,24	11,07	12,83	15,08	16,75	20,51
6	10,64	12,59	14,45	16,81	18,55	22,46
7	12,02	14,07	16,01	18,48	20,28	24,32

Число свободных степеней ν	q, ε (q=ε) / ε [q=(1+ε)/2]					
	0,9 / 0,8	0,95 / 0,9	0,975 / 0,95	0,99 / 0,98	0,995 / 0,99	0,999 / 0,998
	α (q=1−α/2) / α (q=1−α)					
	0,2 / 0,1	0,1 / 0,05	0,05 / 0,025	0,02 / 0,01	0,01 / 0,005	0,02 / 0,001
8	13,36	15,51	17,53	20,09	21,95	26,12
9	14,68	16,92	19,02	21,67	23,59	27,88
10	15,99	18,31	20,48	23,21	25,19	29,59
11	17,28	19,68	21,92	24,72	26,76	31,26
12	18,55	21,03	23,34	26,22	28,30	32,91
13	19,81	22,36	24,74	27,69	29,82	34,53
14	21,06	23,68	26,12	29,14	31,32	36,12
15	22,31	25,00	27,49	30,58	32,80	37,70
16	23,54	26,30	28,85	32,00	34,27	39,25
17	24,77	27,59	30,19	33,41	35,72	40,79
18	25,99	28,87	31,53	34,81	37,16	42,31
19	27,20	30,14	32,85	36,19	38,58	43,82
20	28,41	31,41	34,17	37,57	40,00	45,31
21	29,62	32,67	35,48	38,93	41,40	46,80
22	30,81	33,92	36,78	40,29	42,80	48,27
23	32,01	35,17	38,08	41,64	44,18	49,73
24	33,20	36,42	39,36	42,98	45,56	51,18
25	34,38	37,65	40,65	44,31	46,93	52,62
26	35,56	38,89	41,92	45,64	48,29	54,05
27	36,74	40,11	43,19	46,96	49,64	55,48
28	37,92	41,34	44,46	48,28	50,99	56,89
29	39,09	42,56	45,72	49,59	52,34	58,30

Число свободных степеней ν	q, ε (q = ε) / ε [q= (1+ε)/2]					
	0,9 / 0,8	0,95 / 0,9	0,975 / 0,95	0,99 / 0,98	0,995 / 0,99	0,999 / 0,998
	α (q = 1−α/2) / α (q = 1− α)					
	0,2 / 0,1	0,1 / 0,05	0,05 / 0,025	0,02 / 0,01	0,01 / 0,005	0,02 / 0,001
30	40,26	43,77	46,98	50,98	53,67	59,70
34	44,90	48,60	51,97	56,06	58,96	65,25
36	47,21	51,00	54,44	58,62	61,58	67,99
38	49,51	53,38	56,90	61,16	64,18	70,70
40	51,81	55,76	59,34	63,69	66,77	73,40
42	54,09	58,12	61,78	66,21	69,34	76,08
44	56,37	60,48	64,20	68,71	71,89	78,75
46	58,64	62,83	66,62	71,20	74,44	81,40
48	60,91	65,17	69,02	73,68	76,97	84,04
50	63,17	67,50	71,42	76,15	79,49	86,66
55	68,80	73,31	77,38	82,29	85,75	93,17
60	74,40	79,08	83,30	88,38	91,95	99,61
65	79,97	84,82	89,18	94,42	98,11	105,99
70	85,53	90,53	95,02	100,43	104,21	112,32
75	91,06	96,22	100,84	106,39	110,29	118,60
80	96,58	101,88	106,63	112,33	116,32	124,84
85	102,08	107,52	112,39	118,24	122,32	131,04
90	107,56	113,15	118,14	124,12	128,30	137,21
95	113,04	118,75	123,86	129,97	134,25	143,34
100	118,50	124,56	129,56	135,81	140,17	149,45

Таблица П.8

Квантиль $F_{\nu_1\nu_2\,q}$ F-распределения при $q=0{,}95$; $\alpha=0{,}1$ $(q=1-\alpha/2)$; $\alpha=0{,}05$ $(q=1-\alpha)$; $\varepsilon=0{,}9$ $(q=(1+\varepsilon)/2)$; $\varepsilon=0{,}95$ $(q=\varepsilon)$

ν_2	ν_1: 1	2	3	4	5	6	7	8
1	161,4	199,5	215,7	224,6	230,2	234,0	236,8	238,9
2	18,51	19,00	19,16	19,25	19,30	19,33	19,35	19,37
3	10,13	9,55	9,28	9,12	0,01	8,94	8,89	8,85
4	7,71	6,94	6,59	6,39	6,26	6,16	6,09	6,04
5	6,61	5,79	5,41	5,19	5,05	4,95	4,88	4,82
6	5,99	5,14	4,76	4,53	4,39	4,28	4,21	4,15
7	5,59	4,74	4,35	4,12	3,97	3,87	3,79	3,73
8	5,32	4,46	4,07	3,84	3,69	3,58	3,50	3,44
9	5,12	4,26	3,86	3,63	3,48	3,37	3,29	3,23
10	4,96	4,10	3,71	3,48	3,33	3,22	3,14	3,07
12	4,75	3,89	3,49	3,26	3,11	3,00	2,91	2,85
15	4,54	3,58	3,29	3,06	2,90	2,79	2,71	2,64
20	4,35	3,49	3,10	2,87	2,71	2,60	2,51	2,45
25	4,24	3,39	2,99	2,76	2,60	2,49	2,40	2,34
30	4,17	3,32	2,92	2,69	2,53	2,42	2,33	2,27
40	4,08	3,23	2,84	2,61	2,45	2,34	2,25	2,18
60	4,00	3,15	2,76	2,53	2,37	2,25	2,17	2,10
120	3,92	3,07	3,68	2,45	2,29	2,17	2,09	2,02
∞	3,84	3,00	2,60	2,37	2,21	2,10	2,01	1,94

ν_2	ν_1: 9	10	12	15	20	30	40	60	120	∞
1	240,5	241,9	243,9	245,9	248,0	250,1	251,1	252,2	253,3	254,3
2	19,38	19,40	19,41	19,43	19,45	19,46	19,47	19,48	19,49	19,50
3	8,81	8,79	8,74	8,70	8,66	8,62	8,59	8,57	8,55	8,53
4	6,00	5,96	5,91	5,86	5,80	5,75	5,72	5,69	5,66	5,63
5	4,77	4,74	4,68	4,62	4,56	4,50	4,46	4,43	4,40	4,36
6	4,10	4,06	4,00	3,94	3,87	3,81	3,77	3,74	3,70	3,67
7	3,68	3,64	3,57	3,51	3,44	3,38	3,34	3,30	3,27	3,23
8	3,39	3,35	3,28	3,22	3,15	3,08	3,04	3,01	2,97	2,93
9	3,18	3,14	3,07	3,01	2,94	2,86	2,83	2,79	2,75	2,71
10	3,02	2,98	2,91	2,85	2,77	2,70	2,66	2,62	2,58	2,54
12	2,80	2,75	2,69	2,62	2,54	2,47	2,43	2,38	2,34	2,30
15	2,59	2,54	2,48	2,40	2,33	2,25	2,20	2,16	2,11	2,07
20	2,39	2,35	2,28	2,20	2,12	2,04	1,99	1,95	1,90	1,84
25	2,28	2,24	2,16	2,09	2,01	1,92	1,87	1,82	1,77	1,71
30	2,21	2,16	2,09	2,01	1,93	1,84	1,79	1,74	1,68	1,62
40	2,12	2,08	2,00	1,92	1,84	1,74	1,69	1,64	1,58	1,51
60	2,04	1,99	1,92	1,84	1,75	1,65	1,59	1,53	1,47	1,39
120	1,96	1,91	1,83	1,75	1,66	1,55	1,50	1,43	1,35	1,25
∞	1,88	1,83	1,75	1,67	1,57	1,46	1,39	1,32	1,22	1,00

Таблица П.9

Квантиль $F_{\nu_1\nu_2\,q}$ F-распределения при $q=0{,}975$; $\alpha=0{,}05$ $(q=1-\alpha/2)$; $\alpha=0{,}025$ $(q=1-\alpha)$; $\varepsilon=0{,}95$ $(q=(1+\varepsilon/2)$; $\varepsilon=0{,}975$ $(q=\varepsilon)$

ν_2	ν_1: 1	2	3	4	5	6	7	8
1	647,8	799,5	864,2	899,6	921,8	937,1	948,2	956,7
2	38,51	39,00	39,17	39,25	39,30	39,33	39,36	39,37
3	17,44	16,04	15,44	15,10	14,88	14,73	14,62	14,54
4	12,22	10,65	9,98	9,60	9,36	9,20	9,07	8,98
5	10,01	8,43	7,76	7,39	7,15	6,98	6,85	6,76
6	8,81	7,26	6,60	6,23	5,99	5,82	5,70	5,60
7	8,07	6,54	5,89	5,52	5,29	5,12	4,99	4,90
8	7,57	6,06	5,42	5,05	4,82	4,65	4,53	4,43
9	7,21	5,71	5,08	4,72	4,48	4,32	4,20	4,10
10	6,94	5,46	4,83	4,47	4,24	4,07	3,95	3,85
12	6,55	5,10	4,47	4,12	3,89	3,73	3,61	3,51
15	6,20	4,77	4,15	3,80	3,58	3,41	3,29	3,20
20	5,87	4,46	3,86	3,51	3,29	3,13	3,01	2,91
25	5,69	4,29	3,69	3,35	3,13	2,97	2,85	2,75
30	5,57	4,18	3,59	3,25	3,03	3,87	2,75	2,65
40	5,42	4,05	3,46	3,13	2,90	2,74	2,62	2,53
60	5,29	3,93	3,34	3,01	2,79	2,63	2,51	2,41
120	5,15	3,80	3,23	2,89	2,67	2,52	2,39	2,30
∞	5,02	3,69	3,12	2,79	2,57	2,41	2,29	2,19

ν_2	ν_1: 9	10	12	15	20	30	40	60	120	∞
1	963,3	968,6	976,7	984,9	993,1	1001	1006	1010	1014	1018
2	39,39	39,40	39,41	39,43	39,45	39,46	39,47	39,48	39,49	39,50
3	14,47	14,42	14,34	14,25	14,17	14,08	14,04	13,99	13,95	13,90
4	8,90	8,84	8,75	8,66	8,56	8,46	8,41	8,36	8,31	8,26
5	6,68	6,62	6,52	6,43	6,33	6,23	6,18	6,12	6,07	6,02
6	5,52	5,46	5,37	5,27	5,17	5,07	5,01	4,96	4,90	4,85
7	4,82	4,76	4,67	4,57	4,47	4,36	4,31	4,25	4,20	4,14
8	4,36	4,30	4,20	4,10	4,00	3,89	3,84	3,78	3,73	3,67
9	4,03	3,96	3,87	3,77	3,67	3,56	3,51	3,45	3,39	3,33
10	3,78	3,72	3,62	3,52	3,42	3,31	3,26	3,20	3,14	3,08
12	3,44	3,37	3,28	3,18	3,07	2,96	2,91	2,85	2,79	2,72
15	3,12	3,06	2,96	2,86	2,76	2,64	2,59	2,52	2,46	2,40
20	2,84	2,77	2,68	2,57	2,46	2,35	2,29	2,22	2,16	2,09
25	2,68	2,61	2,51	2,41	2,30	2,18	2,12	2,05	1,98	1,91
30	2,57	2,51	2,41	2,31	2,20	2,07	2,01	1,94	1,87	1,79
40	2,45	2,39	2,29	2,18	2,07	1,94	1,88	1,80	1,72	1,64
60	2,33	2,27	2,17	2,06	1,94	1,82	1,74	1,67	1,58	1,48
120	2,22	2,16	2,05	1,94	1,82	1,69	1,61	1,53	1,43	1,31
∞	2,11	2,05	1,94	1,83	1,71	1,57	1,48	1,39	1,27	1,00

Таблица П.10

Квантиль $F_{\nu_1\nu_2 q}$ F-распределения при $q=0,99$; $\alpha=0,02$ $(q=1-\alpha/2)$; $\alpha=0,01$ $(q=1-\alpha)$; $\varepsilon=0,98$ $(q=(1+\varepsilon)/2)$; $\varepsilon=0,99$ $(q=\varepsilon)$

ν_2	ν_1 = 1	2	3	4	5	6	7	8	9	10	12	15	20	30	40	60	120	∞
1	4052	5000	5403	5625	5764	5859	5928	5982	6022	6056	6106	6157	6209	6261	6287	6313	6339	6366
2	98,50	99,00	99,17	99,25	99,30	99,33	99,36	99,37	99,39	99,40	99,42	99,43	99,45	99,47	99,47	99,48	99,49	99,50
3	34,12	30,82	29,46	28,71	28,24	27,91	27,67	27,49	27,35	27,23	27,05	26,87	26,69	26,50	26,41	26,32	26,22	26,13
4	21,20	18,00	16,69	15,98	15,52	15,21	14,98	14,80	14,66	14,55	14,37	14,20	14,02	13,84	13,75	13,65	13,56	13,46
5	16,26	13,27	12,06	11,39	10,97	10,67	10,46	10,29	10,16	10,05	9,89	9,72	9,55	9,38	9,29	9,20	9,11	9,02
6	13,75	10,92	9,78	9,15	8,75	8,47	8,26	8,10	7,98	7,87	7,72	7,56	7,40	7,23	7,14	7,06	6,97	6,88
7	12,25	9,55	8,45	7,85	7,46	7,19	6,99	6,84	6,72	6,62	6,47	6,31	6,16	5,99	5,91	5,82	5,74	5,65
8	11,26	8,65	7,59	7,01	6,63	6,37	6,18	6,03	5,91	5,81	5,67	5,52	5,36	5,20	5,12	5,03	4,95	4,86
9	10,56	8,02	6,99	6,42	6,06	5,80	5,61	5,47	5,35	5,26	5,11	4,96	4,81	4,65	4,57	4,48	4,40	4,31
10	10,04	7,56	6,55	5,99	5,64	5,39	5,20	5,06	4,94	4,85	4,71	4,56	4,41	4,25	4,17	4,08	4,00	3,91
12	9,33	6,93	5,95	5,41	5,06	4,82	4,64	4,50	4,39	4,30	4,16	4,01	3,86	3,70	3,62	3,54	3,45	3,36
15	8,68	6,36	5,42	4,89	4,56	4,32	4,14	4,00	3,89	3,80	3,67	3,52	3,37	3,21	3,13	3,05	2,96	2,87
20	8,10	5,85	4,94	4,43	4,10	3,87	3,70	3,56	3,46	3,37	3,23	3,09	2,94	2,76	2,69	2,61	2,52	2,42
25	7,77	5,57	4,68	4,18	3,85	3,63	3,46	3,32	3,22	3,13	2,99	2,85	2,70	2,54	2,45	2,36	2,27	2,17
30	7,56	5,39	4,51	4,02	3,70	3,47	3,30	3,17	3,07	2,98	2,84	2,70	2,55	2,39	2,30	2,21	2,11	2,01
40	7,31	5,18	4,31	3,83	3,51	3,29	3,12	2,99	2,89	2,80	2,66	2,52	2,37	2,20	2,11	2,02	1,92	1,80
60	7,08	4,98	4,13	3,65	3,34	3,12	2,95	2,82	2,72	2,63	2,50	2,35	2,20	2,03	1,94	1,84	1,73	1,60
120	6,85	4,79	3,95	3,48	3,17	2,96	2,79	2,66	2,56	2,47	2,34	2,19	2,03	1,86	1,76	1,66	1,53	1,38
∞	6,63	4,61	3,78	3,32	3,02	2,80	2,64	2,51	2,41	2,32	2,18	2,04	1,88	1,70	1,59	1,47	1,32	1,00

Таблица П.11

Квантиль $F_{\nu_1\nu_2 q}$ F-распределения при $q=0,999$; $\alpha=0,02$ $(q=1-\alpha/2)$; $\alpha=0,001$ $(q=1-\alpha)$; $\varepsilon=0,998$ $(q=(1+\varepsilon)/2)$; $\varepsilon=0,999$ $(q=\varepsilon)$

ν_2	ν_1 = 1	2	3	4	5	6	7	8	9	10	12	15	20	30	40	60	120	∞
1	4053*	5000*	5404*	5625*	5764*	5859*	5929*	5981*	6023*	6056*	6107*	6158*	6209*	6261*	6287*	6313*	6340*	6366*
2	998,5	999,0	999,2	999,2	999,3	999,3	999,4	999,4	999,4	999,4	999,4	999,4	999,4	999,5	999,5	999,5	999,5	999,5
3	167,0	148,5	141,1	137,1	134,6	132,8	131,6	130,6	129,9	129,2	128,3	127,4	126,4	125,4	125,0	124,5	124,0	123,5
4	74,14	61,25	56,18	53,44	51,71	50,53	49,66	49,00	48,47	48,05	47,41	46,76	46,10	45,43	45,09	44,75	44,40	44,05
5	47,18	37,12	33,20	31,09	29,75	28,84	28,16	27,64	27,24	26,92	26,42	25,91	25,39	24,87	24,60	24,33	24,06	23,79
6	35,51	27,00	23,70	21,92	20,81	20,03	19,46	19,03	18,69	18,41	17,99	17,56	17,12	16,67	16,44	16,21	15,99	15,75
7	29,25	21,69	18,77	17,19	16,21	15,52	15,02	14,63	14,33	14,08	13,71	13,32	12,93	12,53	12,33	12,12	11,91	11,70
8	25,42	18,49	15,83	14,39	13,49	12,86	12,40	12,04	11,77	11,54	11,19	10,84	10,48	10,11	9,92	9,73	9,53	9,33
9	22,86	16,39	13,90	12,56	11,71	11,13	10,70	10,37	10,11	9,89	9,57	9,24	8,90	8,55	8,37	8,19	8,00	7,81
10	21,04	14,91	12,55	11,28	10,48	9,92	9,52	9,20	8,96	9,75	8,45	8,13	7,80	7,47	7,30	7,12	6,94	6,76
12	18,64	12,97	10,80	9,63	8,89	8,38	8,00	7,71	7,48	7,29	7,00	6,71	6,40	6,09	5,93	5,76	5,59	5,42
15	16,59	11,34	9,34	8,25	7,57	7,09	6,74	6,47	6,26	6,08	5,81	5,54	5,25	4,95	4,80	4,64	4,47	4,31
20	14,82	9,95	8,10	7,10	6,46	6,02	5,69	5,44	5,24	5,08	4,82	4,56	4,29	4,00	3,86	3,70	3,54	3,38
25	13,88	9,22	7,45	6,49	5,88	5,46	5,15	4,91	4,71	4,56	4,31	4,06	3,79	3,52	3,37	3,22	3,06	2,89
30	11,29	8,77	7,05	6,12	5,53	5,12	4,82	4,58	4,39	4,24	4,00	3,75	3,49	3,22	3,07	2,92	2,76	2,59
40	12,61	8,25	6,60	5,70	5,13	4,73	4,44	4,21	4,02	3,87	3,64	3,40	3,15	2,87	2,73	2,57	2,41	2,23
60	11,97	7,76	6,17	5,31	4,76	4,37	4,09	3,87	3,69	3,54	3,31	3,08	2,83	2,55	2,41	2,25	2,08	1,89
120	11,38	7,32	5,79	4,95	4,42	4,04	3,77	3,55	3,38	3,24	3,02	2,78	2,53	2,26	2,11	1,95	1,76	1,54
∞	10,83	6,91	5,42	4,62	4,10	3,74	3,47	3,27	3,10	2,96	2,74	2,51	2,27	1,99	1,84	1,66	1,45	1,00

* Значения умножаются на 100.

Квантиль $C_{l\nu\alpha}$ порядка α распределения Кокрена $C_{l\nu}$ с $(l,\ \nu)$ степенями свободы при $\alpha=0{,}05$

l	ν						
	1	2	3	4	5	6	7
2	0,9985	0,9750	0,9392	0,9057	0,8772	0,8534	0,8332
3	0,9669	0,8709	0,7977	0,7457	0,7071	0,6771	0,6530
4	0,9065	0,7679	0,6841	0,6287	0,5895	0,5598	0,5365
5	0,8412	0,6838	0,5981	0,5440	0,5063	0,4783	0,4564
6	0,7808	0,6161	0,5321	0,4803	0,4447	0,4184	0,3980
7	0,7271	0,5612	0,4800	0,4307	0,3974	0,3726	0,3535
8	0,6798	0,5157	0,4377	0,3910	0,3595	0,3362	0,3185
9	0,6385	0,4775	0,4027	0,3584	0,3286	0,3067	0,2901
10	0,6020	0,4450	0,3733	0,3311	0,3029	0,2823	0,2666
12	0,5410	0,3924	0,3264	0,2880	0,2624	0,2439	0,2299
15	0,4709	0,3346	0,2758	0,2419	0,2195	0,2034	0,1911
20	0,3894	0,2705	0,2205	0,1921	0,1735	0,1602	0,1501
24	0,3434	0,2354	0,1907	0,1656	0,1493	0,1374	0,1286
30	0,2929	0,1980	0,1593	0,1377	0,1237	0,1137	0,1061
40	0,2370	0,1576	0,1259	0,1082	0,0968	0,0887	0,0827
60	0,1737	0,1131	0,0895	0,0765	0,0682	0,0623	0,0583
120	0,0998	0,0632	0,0495	0,0419	0,0371	0,0337	0,0312
∞	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

l	ν						
	8	9	10	16	36	144	∞
2	0,8159	0,8010	0,7880	0,7341	0,6602	0,5813	0,5000
3	0,6333	0,6167	0,6025	0,5466	0,4748	0,4031	0,3333
4	0,5175	0,5017	0,4884	0,4366	0,3720	0,3093	0,3500
5	0,4387	0,4241	0,4118	0,3645	0,3066	0,2513	0,2000
6	0,3817	0,3682	0,3568	0,3135	0,2612	0,2119	0,1667
7	0,3384	0,3259	0,3154	0,2756	0,2278	0,1833	0,1429
8	0,3043	0,2926	0,2829	0,2462	0,2022	0,1616	0,1250
9	0,2768	0,2659	0,2568	0,2226	0,1820	0,1446	0,1111
10	0,2541	0,2439	0,2353	0,2032	0,1655	0,1308	0,1000
12	0,2187	0,2098	0,2020	0,1737	0,1403	0,1100	0,0833
15	0,1815	0,1736	0,1671	0,1429	0,1144	0,0889	0,0667
20	0,1422	0,1357	0,1303	0,1108	0,0879	0,0675	0,0500
24	0,1216	0,1160	0,1113	0,0942	0,0743	0,0567	0,0417
30	0,1002	0,0958	0,0921	0,0771	0,0604	0,0457	0,0333
40	0,0780	0,0745	0,0713	0,0595	0,0462	0,0347	0,0250
60	0,0552	0,0520	0,0497	0,0411	0,0316	0,0234	0,0167
120	0,0292	0,0279	0,0266	0,0218	0,0165	0,0120	0,0083
∞	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000

СПИСОК ЛИТЕРАТУРЫ

1. **Абрамов С. А.** Элементы программирования. — М.: Наука, 1982.
2. **Бедрековский М. А., Волга В. В., Кручинкин Н. С.** Микропроцессоры. — М.: Радио и связь, 1981.
3. **Блох А. Ш., Павловский А. И., Пенкрат В. В.** Программирование на микрокалькуляторах. — Минск, Вышэйшая школа, 1981.
4. **Гильде В., Альтрихтер З.** С микрокалькулятором в руках: Пер. с нем. — М.: Мир, 1980.
5. **Гмурман В. Е.** Руководство к решению задач по теории вероятностей и математической статистике. — М.: Высшая школа, 1975.
6. **Джонсон Н., Лион Ф.** Статистика и планирование эксперимента в технике и науке. Методы обработки данных: Пер. с англ. — М.: Наука, 1980.
7. **Дроздов Е. А., Комарницкий В. А., Пятибратов А. П.** Электронные вычислительные машины единой системы. — М.: Машиностроение, 1981.
8. **Дьяконов В. П.** Расчет нелинейных импульсных устройств на программируемых микрокалькуляторах: Справочное пособие. — М. — Радио и связь, 1984.
9. **Ершов А. П.** Введение в теоретическое программирование. — М.: Наука, 1977.
10. **Калабеков Б. А.** Применение ЭВМ в инженерных расчетах в технике связи. — М.: Радио и связь, 1981.
11. **Карамзин Л. Н., Чистов Э. А.** Таблицы функций Бесселя от мнимого аргумента и интегралов от них. — М.: Наука, 1958.
12. **Кендэл М.** Временные ряды: Пер. с англ. — М.: Финансы и статистика. 1981.
13. **Копченова Н. В., Марон И. А.** Вычислительная математика в примерах и задачах. — М.: Наука, 1972.
14. **Королев Л. Н.** Структуры ЭВМ и их математическое обеспечение. — М.: Наука, 1978.
15. **Кройль Г.** Что умеет мой микрокалькулятор. Пер. с нем. — М.: Мир, 1981.
16. **Людмен А. А.** Меньше программных шагов//Электроника. 1982. № 5.
17. **Мак-Кракен Д., Дорн У.** Численные методы и программирование на ФОРТРАНе: Пер. с англ. — М.: Мир, 1977.
18. **Микрокалькуляторы.** Технические и конструктивные характеристики./Кузнецов Е. Ю., Острецов Б. В., Минкин Л. К., Егорова Ю. И. — М.: Радио и связь, 1984.

19. **Мюллер П., Нойман П., Шторм Р.** Таблицы по математической статистике: Пер. с нем. — М.: Финансы и статистика, 1982.
20. **Мясников В. А., Майоров С. А., Новиков Г. И.** ЭВМ для всех. — М.: Знание, 1980.
21. **Поллард Дж.** Справочник по вычислительным методам статистики: Пер. с англ. — М.: Финансы и статистика, 1982.
22. **Прокофьев В. А.** Программирование для мини-ЭВМ. — М.: Сов. радио, 1979.
23. **Постников М. М.** Устойчивые многочлены. — М.: Наука, 1981.
24. **Самарский А.** Современная прикладная математика и вычислительный эксперимент//Коммунист. 1983. № 18.
25. **Сергеев Н. П., Домнин Л. Н.** Алгоритмизация и программирование. — М.: Радио и связь, 1982.
26. **Справочник** по специальным функциям с формулами, графиками и таблицами/Под ред. М. Абрамовица и И. Стиган: Пер. с англ. — М.: Наука, 1979.
27. **Трохименко Я. К.** Искусство программирования программируемых микрокалькуляторов//Известия вузов. Радиоэлектроника. 1984. Т. 27. № 2, 6, 9, 10, 11.
28. **Трохименко Я. К., Любич Ф. Д.** Инженерные расчеты на микрокалькуляторах. — Киев: Техника, 1980.
29. **Трохименко Я. К., Любич Ф. Д.** Искусство программирования программируемых микрокалькуляторов//Известия вузов. Радиоэлектроника. 1983. Т. 26. № 6, 7, 8, 9, 10, 11, 12.
30. **Трохименко Я. К., Любич Ф. Д.** Искусство программирования программируемых микрокалькуляторов// Известия вузов, Радиотехника. 1984. Т. 27. № 1, 3, 4, 5.
31. **Трохименко Я. К., Любич Ф. Д.** Радиотехнические расчеты на микрокалькуляторах. Справочное пособие. — М.: Радио и связь, 1983.
32. **Трохименко Я. К., Любич Ф. Д.** Цифровое моделирование на микрокалькуляторах//Известия вузов. Радиоэлектроника. 1979. Т. 23, № 6.
33. **Финк Л. М.** Целочисленные операции с помощью программируемых микрокалькуляторов// Известия вузов. Радиоэлектроника. Т. 27. № 7.
34. **Форсайт Дж., Малькольм М., Моулер К.** Машинные методы математических вычислений: Пер. с англ. — М.: Мир, 1980.
35. **Хэмминг Р. В.** Цифровые фильтры: Пер. с англ. — М.: Сов. радио, 1980.
36. **Цветков А. Н.** Искусство программирования программируемых микрокалькуляторов. Анализ и интерпретация экспериментальных данных//Известия вузов. Радиоэлектроника. 1984. Т. 27. № 8.
37. **Цветков А. Н.** Прикладные программы для микро-ЭВМ «Электроника Б3-21». — М.: Финансы и статистика, 1982.
38. **Цветков А. Н., Епанечников В. А.** Прикладные программы для микро-ЭВМ. — М.: Финансы и статистика, 1984.
39. **Цимринг Ш. Е.** Специальные функции: Программы для микрокалькулятора «Электроника Б3-21». — М.: Радио и связь, 1983.
40. **Чакань А.** Что умеет карманная ЭВМ?: Пер. с венг. — М.: Радио и связь, 1982.
41. **Шуп Т.** Решение инженерных задач на ЭВМ: Практическое руководство: Пер. с англ. — М.: Мир, 1982.

ОГЛАВЛЕНИЕ

РАЗДЕЛ ВТОРОЙ

МЕТОДЫ, АЛГОРИТМЫ И ПРОГРАММЫ РЕШЕНИЯ ЗАДАЧ НА МИКРОКАЛЬКУЛЯТОРАХ