

В. П. Дьяконов

СПРАВОЧНИК
по алгоритмам
и программам
на языке бейсик
для персональных
ЭВМ



В. П. ДЬЯКОНОВ

СПРАВОЧНИК по алгоритмам и программам на языке бейсик для персональных ЭВМ



МОСКВА «НАУКА»

ГЛАВНАЯ РЕДАКЦИЯ

ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

1983

ББК 22.18
Д93
УДК 519.6(083)

Дьяконов В. П. **Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ:** Справочник.— М.: Наука. Гл. ред. физ.-мат. лит., 1989.— 240 с.— ISBN 5-02-014530-0.

Даны краткие сведения о современных отечественных и зарубежных микроЭВМ индивидуального пользования. Описаны основные версии бейсика, наиболее распространенного языка программирования персональных ЭВМ (ПЭВМ), отечественных микро- и миниЭВМ (Электроника-60, Электроника-ДЗ-28, Искра-226, диалоговых вычислительных комплексов ДВК-1, ДВК-2 и др.). Изложены основы программирования на бейсике. Основное внимание уделено общему математическому, алгоритмическому и программному обеспечению расчетов на ПЭВМ. В справочник включена обширная библиотека прикладных программ на бейсике (более 300 программ), обеспечивающих реализацию основных численных методов, вычисление большинства специальных функций и решение ряда практических задач в различных областях науки и техники.

Для инженеров, научно-технических работников и студентов вузов.

Библиогр. 43 назв.

Рецензент доктор технических наук *С. В. Черемных*

ДЬЯКОНОВ Владимир Павлович

**СПРАВОЧНИК ПО АЛГОРИТМАМ И ПРОГРАММАМ НА ЯЗЫКЕ БЕЙСИК
ДЛЯ ПЕРСОНАЛЬНЫХ ЭВМ**

Редактор *Л. Г. Силкова*

Художественный редактор *Г. М. Коровина*

Технический редактор *С. Я. Шкляр*. Корректоры *Е. Ю. Рычагова, М. Л. Медведская*

ИБ № 32350

Подписано к печати 23.12.88. Т-23076. Формат 70×100/16. Бумага книжно-журнальная. Гарнитура литературная. Печать офсетная. Усл. печ. л. 19,5. Усл. кр.-отт. 39,33. Уч.-изд. л. 25,16. Допечатка тиража 100 000 экз. Заказ 1274. Цена 1 р. 40 к.

Ордена Трудового Красного Знамени издательство «Наука».

Главная редакция физико-математической литературы.

117071 Москва В-71, Ленинский проспект, 15.

Ленинградская типография № 2 головное предприятие ордена Трудового Красного Знамени Ленинградского объединения «Техническая книга» им. Евгении Соколовой Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли.

198052, Ленинград, Л-52, Измайловский проспект, 29.

Д $\frac{1702070000-021}{053(02)-89}$ 48-87

ISBN 5-02-014530-0

© Издательство «Наука»
Главная редакция
физико-математической
литературы, 1987

Предисловие	5
Как пользоваться справочником	7
Глава 1. Основные характеристики и возможности персональных ЭВМ	9
§ 1.1. Современные типы персональных ЭВМ и их возможности	9
§ 1.2. Карманные персональные ЭВМ (Pocket Computers)	11
§ 1.3. Персональные ЭВМ среднего класса (Home Computers)	13
§ 1.4. Профессиональные ЭВМ и вычислительные микросистемы индивидуального пользования	16
§ 1.5. Периферийное оборудование персональных ЭВМ	19
Глава 2. Бейсик — основной язык программирования персональных ЭВМ	21
§ 2.1. Алфавит и основные операторы языка бейсик	21
§ 2.2. Модификации языка бейсик	27
§ 2.3. Арифметические и алгебраические операции, работа в режиме калькулятора	41
§ 2.4. Элементарное программирование на языке бейсик	43
§ 2.5. Специальные вопросы программирования на языке бейсик	48
§ 2.6. Перевод программ с одной версии языка бейсик на другую	51
Глава 3. Алгоритмы и программы элементарных вычислений	54
§ 3.1. Операции с действительными числами	54
§ 3.2. Операции и функции с комплексными числами и переменными	56
§ 3.3. Вычисление степенных многочленов и дробно-рациональных функций	61
§ 3.4. Вычисление ортогональных многочленов	63
§ 3.5. Операции с матрицами	65
§ 3.6. Вычисление факториалов и комбинаторика	70
§ 3.7. Преобразования координат и векторный анализ	71
Глава 4. Алгоритмы и программы реализации основных численных методов	75
§ 4.1. Решение систем линейных уравнений	75
§ 4.2. Интерполяция и экстраполяция	78
§ 4.3. Решение нелинейных и трансцендентных уравнений	86
§ 4.4. Решение систем нелинейных уравнений	91
§ 4.5. Решение алгебраических уравнений с действительными и комплексными коэффициентами	93
§ 4.6. Поиск экстремумов функций одной и множества переменных	96
§ 4.7. Численное дифференцирование и вычисление коэффициентов чувствительности	100
§ 4.8. Вычисление определенных интегралов	104
§ 4.9. Вычисление определенных интегралов специального вида	108
§ 4.10. Решение систем дифференциальных уравнений	111
§ 4.11. Гармонический синтез	116
§ 4.12. Вычисление собственных значений и векторов матриц	117
Глава 5. Спектральный, статистический, корреляционный и регрессионный анализ	122
§ 5.1. Спектральный анализ на основе дискретного преобразования Фурье	122
§ 5.2. Специальные виды спектрального анализа	128
§ 5.3. Статистический анализ и подготовка гистограмм	133
§ 5.4. Реализация метода Монте-Карло	136
§ 5.5. Корреляционный анализ	137
§ 5.6. Регрессионный анализ (приближение функций по методу наименьших квадратов)	138
§ 5.7. Сглаживание данных эксперимента	144

Глава 6. Вычисление специальных функций	147
§ 6.1. Методы вычисления специальных функций	147
§ 6.2. Интегральные показательные функции	147
§ 6.3. Интегральные синус и косинус	148
§ 6.4. Гамма-функции (включая неполные)	149
§ 6.5. Функции Бесселя (включая модифицированные)	151
§ 6.6. Функции Эйри	153
§ 6.7. Интегралы Френеля	153
§ 6.8. Эллиптические интегралы	154
§ 6.9. Функции Струве, Ангера и Вебера	155
§ 6.10. Гипергеометрические функции	156
§ 6.11. Дилогарифм	156
§ 6.12. Функции Кельвина	157
§ 6.13. Функции Дебая и Зиверта	157
§ 6.14. Интеграл вероятности и родственные ему функции	157
§ 6.15. Некоторые статистические функции	159
Глава 7. Прикладные программы технических и экономических расчетов	162
§ 7.1. Типовые электротехнические расчеты	162
§ 7.2. Расчет индуктивных элементов	166
§ 7.3. Расчет емкостных элементов и конденсаторов	171
§ 7.4. Расчет линий передач и задержки	174
§ 7.5. Расчет усилителей	180
§ 7.6. Расчет активных фильтров	186
§ 7.7. Расчет нелинейных и ключевых электронных устройств	207
§ 7.8. Расчеты в механике и термодинамике	215
§ 7.9. Финансово-экономические расчеты	217
Приложение 1. Подготовка к работе системы подготовки программ на базе микроЭВМ Электроника-ДЗ-28	220
Приложение 2. Номера ошибок и их содержание для систем подготовки программ на базе микроЭВМ Электроника-ДЗ-28	221
Приложение 3. Подготовка ПЭВМ FX-702P к работе	222
Приложение 4. Номера ошибок и их содержание для ПЭВМ FX-702P	223
Приложение 5. Программная реализация некоторых численных методов частного применения	223
§ П5.1. Построение полинома по его действительным корням	223
§ П5.2. Обращение матрицы, вычисление определителя и решение систем линейных уравнений с разными векторами свободных членов	224
§ П5.3. Решение системы линейных уравнений методом отражения	225
§ П5.4. Решение системы линейных уравнений методом простых итераций	225
§ П5.5. Решение системы линейных уравнений методом Зейделя	226
§ П5.6. Решение системы линейных уравнений с переопределенной матрицей	226
§ П5.7. Приближенное вычисление нормального решения системы линейных уравнений с вырожденной матрицей	227
§ П5.8. Решение системы нелинейных уравнений методом простых итераций	228
§ П5.9. Вычисление спектра реакции нелинейной системы с аналитически заданной передаточной характеристикой на гармоническое воздействие	228
§ П5.10. Регрессия для 16 видов парных зависимостей $y(x)$	228
§ П5.11. Сплайн-аппроксимация, интерполяция и экстраполяция	231
§ П5.12. Пакет программ с матричными операторами	232
§ П5.13. Приближение функций по Чебышеву	234
Список литературы	237
Предметный указатель	239

Прогресс в экономике, промышленности, науке и технике, в сфере образования сейчас во многом зависит от массового внедрения вычислительной техники. Не случайно задача повышения «компьютерной грамотности» населения стала государственной и политической задачей. Ее решение возможно благодаря интенсивному развитию нового поколения вычислительных средств — микроЭВМ индивидуального пользования или персональных ЭВМ (ПЭВМ).

Уже сейчас у нас в стране в пользовании находятся многие десятки тысяч таких ЭВМ — от настольных отечественных вычислительных микросистем индивидуального пользования (на базе микроЭВМ Электроника-60, Электроника-ДЗ-28, Электроника-ТЗ-59, Искра-226 и др.) и диалоговых вычислительных комплексов (ДВК-1, ДВК-2, ДВК-3) до зарубежных карманных компьютеров (Pocket Computers FX-702P, PC-1211, PC-1500 и др.). Осваивается производство дешевых и массовых домашних компьютеров (Home Computers) типа Агат, Электроника-БК-0010 и др.

Однако пока даже существующий парк таких ЭВМ используется недостаточно эффективно. Главной причина этого в том, что пользователями ПЭВМ являются в основном непрофессионалы в области вычислительной техники и вычислительной математики. Они слабо знакомы с эксплуатацией ЭВМ, языками и основами их программирования, разнообразными численными методами. Литературы, где все эти вопросы, с учетом специфики персонального использования ПЭВМ, рассматривались бы одновременно, пока практически нет. Многие ПЭВМ поставляются пользователям с весьма краткими (и рассчитанными на профессионалов) описаниями, часто без библиотек прикладных программ. В результате ПЭВМ нередко превращается в «черный ящик» на столе пользователя, таинственные возможности которого остаются нераскрытыми.

Справочник ориентирован на отмеченную выше обширную категорию пользователей ПЭВМ. При его подготовке учтена специфика нынешнего этапа применения ПЭВМ в СССР, т. е. использование ПЭВМ главным образом для автоматизации решения рутинных научно-технических, статистических и экономических задач. Поэтому, а также с учетом ограниченного объема справочника в нем мало внимания уделено игровым задачам, применению ПЭВМ в быту, решению сложных информационных задач (обработка графиков, редактирование текстов и т. д.). Описание таких применений должно быть предметом специальной литературы.

По построению справочник похож на ранее изданную книгу автора [10], посвященную расчетам на программируемых микрокалькуляторах (ПМК). Более того, сохранена значительная часть контрольных примеров [10]. Это, по мнению автора, облегчит массовой категории пользователей ПМК освоение новой, гораздо более мощной вычислительной техники — ПЭВМ.

В справочнике кратко описаны основные типы отечественных и зарубежных ПЭВМ, их периферийное оборудование и основной язык программирования — бейсик (BASIC). Рассмотрен ряд типовых версий бейсика, от простых до расширенных, содержащих команды обработки символьных переменных, косвенной адресации и построения графиков. Даны основы программирования на бейсике и перевода программ с одной версии бейсика на другую.

Основное внимание в справочнике уделено описанию общего прикладного математического, алгоритмического и программного обеспечения ПЭВМ, рассчитанного прежде всего на научно-технические и статистические расчеты. По сравнению с [10] существенно расширено

описание алгоритмов и увеличена сложность решаемых задач. Так, в справочнике описаны универсальные программы для решения систем линейных и дифференциальных уравнений (в том числе с автоматическим выбором шага интегрирования), численного интегрирования с заданной точностью, вычисления всех корней полиномов с действительными и комплексными коэффициентами и т. д. Существенно расширен круг вычисляемых специальных функций. В последней главе дан ряд прикладных программ для решения задач в некоторых конкретных областях науки и техники. Разумеется, эти программы не исчерпывают решения всего многообразия таких задач. При использовании всех возможностей персональных ЭВМ на них можно решать сложные научно-технические задачи, вплоть до проектирования космических аппаратов [29].

Справочник ориентирован на научно-технических работников, инженеров, техников и студентов вузов и техникумов. Поскольку подобное справочное руководство подготовлено впервые, автор отдает себе отчет в том, что книга не лишена недостатков, и с благодарностью примет советы и замечания по ее содержанию. Автор выражает глубокую благодарность рецензенту доктору технических наук, профессору С. В. Черемных, кандидату технических наук, доценту Т. А. Самойловой, Т. А. Калаевой и всем коллегам, оказавшим помощь автору в подготовке программ и рукописи. Пожелания по книге следует направлять по адресу: 117071 Москва В-71, Ленинский просп., 15. Главная редакция физико-математической литературы издательства «Наука».

В. П. Дьяконов

КАК ПОЛЬЗОВАТЬСЯ СПРАВОЧНИКОМ

1. Приведенные в справочнике программы даны на версии языка бейсик умеренной сложности, применяемой в массовых отечественных системах подготовки программ на базе настольной микроЭВМ Электроника-ДЗ-28. Они легко переводятся на версии бейсика, применяемые в диалоговых вычислительных комплексах ДВК-1, ДВК-2 и ДВК-3, микроЭВМ Электроника-60, Искра-226 и др. Для перевода программ рекомендуется изучить материал гл. 2 и описание версии языка бейсик, применяемой в ПЭВМ, которой располагает пользователь.

2. Программы содержат детальный диалог с пользователем, который легко воспринимается при чтении листинга (текста) программы. В первой строке листинга дается подробное наименование программы. Порядок использования программы указывается комментариями, выводимыми на экран дисплея. Поэтому в поясняющем тексте подробное описание правил использования каждой программы не приводится.

3. Все программы рассчитаны на вывод результатов вычислений на экран терминала (дисплея) в экспоненциальной форме, как правило, с одним знаком до десятичной точки и девятью — после нее. Для вывода результатов на печать принтером и изменения формата в программу следует включить команды вида `PRINT #1...` (на место многоточия записываются символы, задающие формат печати).

4. Программы даны в виде распечатки листингов, что исключает возникновение типографских ошибок. Однако после ввода программ в ЭВМ нужно тщательно проверить их как по контрольным примерам, так и тестам пользователя, учитывающим специфику решаемых задач и области применения выбранных численных методов.

5. Между буквенными обозначениями в тексте описания и в программе могут быть легко интерпретируемые различия. Например, буква *e* в программе может обозначаться как *E*, строчная буква *m* — как прописная *M* и т. д. Эти различия обусловлены тем, что в бейсике используются только прописные латинские (а в комментариях и русские) буквы. В отдельных случаях (но не всегда) эти различия оговорены. Как правило, смысл обозначений поясняется комментариями в программах.

6. Следует помнить, особенно при использовании программ, реализующих итерационные методы, что заданная погрешность вычислений ϵ относится к погрешности метода, но не к применяемым исходным формулам! Например, приближенные формулы для инженерного расчета индуктивности или емкости могут иметь погрешность, существенно превосходящую ϵ . Оценка погрешности приближенных и эмпирических формул в ряде случаев не дается, поскольку она входит в сферу профессиональных познаний пользователя.

7. При проведении важных вычислений рекомендуется внимательно разобраться с алгоритмом вычислений и применяемой программой. Следует помнить, что скрытые и неожиданно выявляющиеся ошибки могут быть даже в тщательно отлаженных программах, выдержавших испытания контрольными тестами. Лучший путь исключения таких ошибок заключается в применении программ, полностью понятных пользователю. Естественность и простота записи математических выражений и логических условий на бейсике существенно облегчает разбор программ и способствует осмысленному их применению.

8. В справочнике описаны полностью законченные программы. Однако, удалив строки программ, содержащие операторы ввода исходных данных и вывода результатов, большинство программ можно использовать в качестве подпрограмм в сложных расчетах. Разумеется, при этом может потребоваться изменение номеров строк и адресов.

9. Материал справочника ориентирован на решение расчетных (математических, инженерных и др.) задач общего характера. Поэтому основное внимание уделено алгоритмической ясности программ. В программах по возможности использованы операторы, встречающиеся во всех версиях языка программирования бейсик. Применение некоторых операторов, например DEF FN, FN, ON и др., сознательно ограничено, так как в ряде версий языка бейсик они отсутствуют и могут быть заменены другими операторами. Эти меры направлены на придание программам большей универсальности и на облегчение перевода программ на другие версии бейсика.

10. Наиболее совершенные версии языка бейсик обладают обширными дополнительными возможностями: обработка текстов, графика, средства создания машинных фильмов, звуковые эффекты и т. д. Однако в реализациях этих возможностей у различных версий языка бейсик имеется много принципиальных отличий. Их описание, как отмечалось в предисловии, выводит за рамки данной книги. Однако пользователи ПЭВМ могут легко дополнить описанные в справочнике программы выводом графиков, заданием звуковых эффектов и другими возможностями.

11. Математические выражения и формулы в справочнике даны в окончательном виде без выводов. Читатель, желающий подробно разобраться с математическим обоснованием описанных в справочнике алгоритмов и программ может воспользоваться специальной литературой, включая приведенную в списке литературы справочника.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ И ВОЗМОЖНОСТИ ПЕРСОНАЛЬНЫХ ЭВМ

§ 1.1. Современные типы персональных ЭВМ и их возможности

Персональные ЭВМ (ПЭВМ) являются новым перспективным классом вычислительных средств, ориентированных на использование их одним пользователем. Применение персональных ЭВМ обеспечивает децентрализацию вычислительных процессов, резкое сокращение стоимости машинного времени, повышение удобства эксплуатации и привлечение к работе на ЭВМ обширного круга пользователей, неспециалистов в области вычислительной техники и программирования. Это достигается простотой работы с ПЭВМ, имеющей характер диалога с пользователем.

Карманные персональные ЭВМ (класса Pocket Computers) представляют собой сверхминиатюрные ЭВМ на микросхемах большой степени интеграции (БИС), ориентированные на решение достаточно сложных расчетных и относительно простых информационных задач.

Домашние персональные ЭВМ (класса Home Computers) предназначены для решения расчетных и информационных задач средней сложности, прежде всего относящихся к быту: ведение семейного бюджета, сложные и «интеллектуальные» игры (шахматы, посадка космических кораблей на планеты и т. д.), обучение различным дисциплинам, демонстрация компьютерных слайдов, учебные расчеты и т. д.

Профессиональные персональные ЭВМ ориентированы на решение сложных расчетных и информационных задач различными профессиональными работниками (математиками, физиками, программистами, экономистами и т. д.) в различных, в том числе комфортных; условиях как на работе, так и дома.

Основные свойства персональных ЭВМ (ПЭВМ). ПЭВМ должны обладать следующей совокупностью свойств:

- ориентацией на эксплуатацию одним пользователем, как правило, неспециалистом в области вычислительной техники;

- низкой стоимостью (за рубежом от 50 до 5 000 дол.), обеспечивающей возможность приобретения за счет личных средств пользователя либо средств небольших организаций и учреждений;

- простотой конструкции и высокой надежностью, малыми габаритами и массой, обеспечивающими размещение ПЭВМ на части письменного или рабочего стола;

- работой с обширным периферийным оборудованием (см. § 1.5), включающим устройства внешней памяти, алфавитно-цифровые и графические печатающие устройства (принтеры) и др.;

- программированием хотя бы на одном языке высокого уровня, например бейсике;

- обеспечением общения с пользователем в режиме диалога, в ходе которого ПЭВМ и пользователь обмениваются вопросами и ответами на обычном языке (в письменной, а у наиболее совершенных ЭВМ и в устной форме);

- ориентацией на решение широкого круга как сугубо вычислительных и расчетных задач, так и информационных.

Вычислительные микросистемы индивидуального пользования строятся на основе ПЭВМ или настольных микроЭВМ, оснащенных периферийным оборудованием (как минимум дисплеем, накопителем информации на магнитных дисках или ленте и принтером).

Структура вычислительной микросистемы индивидуального пользования дана на рис. 1.1. Отметим назначение ее узлов.

Шина данных и адресов представляет собой обычный или плоский многожильный кабель, связывающий узлы системы. Через эту шину ЭВМ обменивается с другими узлами цифровыми данными и адресами, по которым эти данные должны поступать.

Центральный процессор (ЦП) — устройство, выполняющее основные логические и арифметические операции. Обычно ЦП ПЭВМ строится на одном или нескольких микропроцессорах (МП) с разрядностью (т. е. числом двоичных разрядов) данных от 4 до 32. В состав ЦП входят также арифметико-логическое устройство управления (УУ) и специальные устройства связи ЦП с периферийным оборудованием через общую шину.

Клавишный пульт — устройство для ввода данных и программ в ПЭВМ, по внешнему виду напоминающее пульт пишущей машинки.

Интерфейс клавиатуры — аппаратные и программные средства сопряжения ЦП с клавишным пультом.

Таймер — устройство для формирования временных последовательностей, задающих порядок работы ПЭВМ.

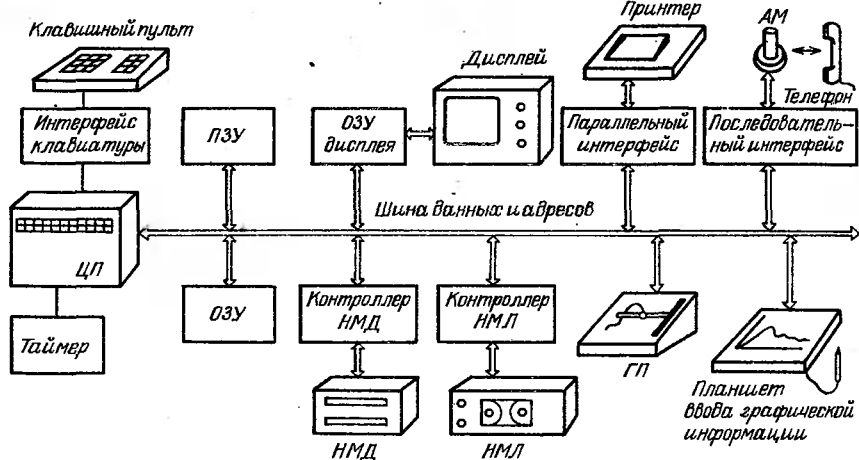


Рис. 1.1. Структура вычислительной микросистемы на основе персональной ЭВМ

Оперативное запоминающее устройство (ОЗУ) — набор БИС (больших интегральных схем), обеспечивающих хранение и временное запоминание цифровой информации. ОЗУ ПЭВМ обычно содержит отделение для хранения информации (одна или несколько страниц), отображаемой дисплеем (ОЗУ дисплея).

Постоянное запоминающее устройство (ПЗУ) служит для постоянного хранения микропрограмм выполнения основных операций ПЭВМ на машинно-ориентированном языке. Обычно ПЗУ содержит и программу-интерпретатор (или транслятор) для программирования ПЭВМ на языке высокого уровня, например бейсике.

Накопитель на магнитном диске (НМД) — устройство для магнитной записи цифровой информации на носитель в виде диска с ферромагнитным покрытием и считывания ее с диска. НМД дополняет ПЗУ ПЭВМ, причем на нем может храниться информация, вводимая пользователем (массивы данных и программы).

Контроллер дискового накопителя (КНМД) — устройство сопряжения ЦП с НМД и контроля работы НМД.

Накопитель на магнитной ленте (НМЛ) — обычный бытовой или специализированный кассетный магнитофон для записи цифровой информации на магнитную ленту. Функции НМЛ аналогичны функциям НМД.

Контроллер накопителя на магнитной ленте (КНМЛ) — устройство сопряжения ЦП с НМЛ и контроля его работы.

Дисплей — устройство отображения алфавитно-цифровой и графической информации. Дисплей обеспечивает контроль вводимой информации, выдачу промежуточных и итоговых результатов работы ПЭВМ, а также контроль информации хода вычислений. Он является основным элементом ПЭВМ, обеспечивающим эффективный диалог с пользователем.

Принтер — устройство печати алфавитно-цифровой (иногда и графической) информации.

Параллельный интерфейс — аппаратные и программные средства сопряжения ПЭВМ с периферийным оборудованием, воспринимающим информацию в виде параллельного кода (например, принтером).

Акустический модем (АМ) — устройство двустороннего обмена информацией между телефонной сетью и ПЭВМ с преобразованием информации в высокочастотные звуковые послыки. АМ используется для связи ПЭВМ с вычислительными и информационными центрами.

Последовательный интерфейс — средства сопряжения ПЭВМ с периферийным оборудованием, воспринимающим информацию в виде последовательного кода (например, АМ).

Графопостроитель (ГП) — устройство для построения графиков и рисунков с высокой разрешающей способностью (иногда функции ГП выполняет универсальный принтер).

Планшет для ввода графической информации — устройство, обеспечивающее преобразование координат точек графика в электрические сигналы, вводимые в ПЭВМ. Иногда его функции выполняет дисплей, снабженный световым пером-датчиком.

Перечисленные узлы полностью входят в состав только наиболее дорогих и сложных ПЭВМ. Более подробные данные о составе ПЭВМ и характеристиках их узлов приведены ниже.

Функциональные возможности ПЭВМ. Области применения ПЭВМ весьма обширны. ПЭВМ могут вести научно-технические, инженерные, экономические и статистические расчеты, облегчают подготовку программ для больших ЭВМ, позволяют решать различные информационные задачи (обработка текстов и графиков, демонстрация подвижных рисунков и иллюстраций к различным дисциплинам, изучаемым пользователем, связь с крупными информационными и вычислительными центрами и представление пользователю имеющихся в них информационных материалов и т. д.). ПЭВМ могут выполнять функции

ЭВМ, управляющих различными приборами: от телевизора, телефона, магнитофона и стиральной машины до сложных контрольно-измерительных комплексов и научного оборудования.

В будущем ПЭВМ приведут к революционным изменениям в информационном обеспечении общества и станут столь же привычным предметом обихода, как телевизор, радиоприемник или магнитофон.

Основные характеристики ПЭВМ: объем памяти ОЗУ и ПЗУ, скорость счета, языки программирования, габариты и масса, состав периферийного оборудования.

Объем памяти ОЗУ и ПЗУ измеряется в килобайтах (1 кбайт = 1024 байт, 1 байт = 8 бит, бит — двоичная единица). Простые ПЭВМ имеют объем памяти ПЗУ и ОЗУ 8 ÷ 16 кбайт. У более сложных объем памяти ОЗУ достигает 128 ÷ 512 кбайт, а объем памяти ПЗУ за счет внешних носителей наращивается до 0,2 ÷ 20 Мбайт (1 Мбайт ≈ 10⁶ байт).

Скорость счета ПЭВМ характеризует число операций в единицу времени. Поскольку видов операций очень много, этот параметр указывается редко. Специалисты судят о скорости счета по примененному в ПЭВМ микропроцессору и тактовой частоте его работы (от 0,5 до 10 МГц).

Пользователей ПЭВМ в первую очередь интересуют функциональные и эксплуатационные возможности ПЭВМ: класс решаемых

задач, габариты и масса, состав периферийного оборудования. Приведенные ниже данные ориентированы на пользователей ПЭВМ, а не на их разработчиков.

§ 1.2. Карманные персональные ЭВМ (Pocket Computers)

Как отмечалось, карманные ПЭВМ класса Pocket Computers представляют собой сверхминиатюрные ЭВМ, выполненные на одной или нескольких БИС. Обычно они снабжены жидкокристаллическим мозаичным индикатором (дисплеем), позволяющим отображать одну или несколько строк текстовых и цифровых данных. Типовые данные ряда ПЭВМ класса Pocket Computers даны в табл. 1.1. В таблицах будут использованы следующие сокращения: КМ — кассетный магнитофон, П — принтер, ГП — графопостроитель, АМ — акустический модем, Д — дисплей, ТД — телевизор-дисплей, ГД — графический дисплей, Т — таймер, НМД — накопитель на магнитном диске, НМЛ — накопитель на магнитной ленте, ТП — термопринтер, прочерк (—) — отсутствующий в описании технический параметр.

Одна из простейших ПЭВМ этого класса РС-1211 (рис. 1.2) имеет клавишный пульс, обеспечивающий побуквенный набор команд языка бейсик (см. гл. 2) [20]. По своим возможностям эта ПЭВМ сопоставима с про-

Таблица 1.1

Характеристики некоторых карманных ПЭВМ класса Pocket Computers

Характеристика	Тип ПЭВМ				
	РС-1210	РС-1211	РС-1500	FX-702P	HP-71B
Фирма и страна-изготовитель	Sharp (Япония)	Sharp (Япония)	Sharp (Япония)	Casio (Япония)	Hewlett Packard (США)
Емкость ПЗУ, кбайт	—	8	16	—	64 + 256
Емкость ОЗУ, кбайт	1	2	3,5 + 16	—	17,5 + 16
Максимальное число шагов программы	400	400 + 1024	—	1680	—
Максимальное число переменных	40	178	—	226	—
Число строк/знаков дисплея	1/24	1/24	1/26	1/20	1/22
Потребляемая мощность, мВт	0,011	0,013	—	0,01	—
Габариты, мм	175 × 71 × 15	175 × 71 × 15	195 × 86 × 22,5	165 × 82 × 17	190 × 98 × 13
Масса, г	163	170	375	176	340
Периферийные устройства	П	КМ, П	КМ, П, ГП, АМ	КМ, П	КМ, П, ГП, АМ, ТД и др.
Язык программирования	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик

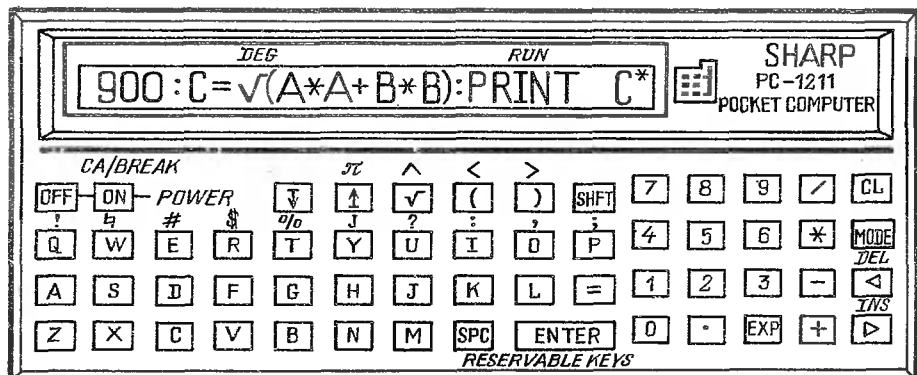


Рис. 1.2. Внешний вид простейшей ПЭВМ класса Pocket Computers PC-1211

граммируемым микрокалькулятором высокого класса [10]. Однако, благодаря программированию на языке бейсик, ПЭВМ PC-1211 обеспечивает проведение вычислений в режиме диалога с пользователем.

ПЭВМ FX-702P выполняет функции сложного непрограммируемого и программируемого микрокалькуляторов для научных и статистических расчетов и ПЭВМ класса Pocket Computers с расширенной версией языка бейсик, содержащей свыше 160 команд. Пульт этой ПЭВМ (рис. 1.3) содержит 65 клавиш [43]. Все клавиши имеют двойное или тройное управление (оно обеспечивается с помощью префиксных клавиш F1 и F2). Наиболее распространенные команды (операторы) бейсика могут вводиться нажатием всего двух клавиш — префиксной и основной с символом нужной команды. Данные о языке бейсик этой ПЭВМ приведены в § 2.2.

ПЭВМ класса Pocket Computers обычно имеют автономное питание от батареек, потребляют ничтожно малую мощность.

Питание для ПЭВМ FX-702P осуществляется от двух литиевых дисковых батарей CR 2032 с напряжением 3 В и емкостью 0,16 А·ч, потребляемая от них мощность не более 0,01 Вт. При нормальной эксплуатации комплекта батарей хватает на 2 года (240 часов непрерывной работы). Предусмотрено автоматическое выключение ПЭВМ через 8 мин, если на ней не ведется работа. Однако при выключении ОЗУ остается подключенным к источнику питания, что обеспечивает долговременную память программ и данных.

В ПЭВМ FX-702P можно вводить до 10 различных программ. Объем памяти этой ПЭВМ составляет 26 переменных (регистров памяти) и 1 680 шагов (каждый оператор бейсика занимает 1 шаг). Число переменных можно увеличить до 226 путем сокращения числа шагов программы до 80. Это обеспечивает формирование одномерных массивов до 200 переменных или двух двумерных массивов 10×10 переменных. Совместно с принтером и бытовым кассетным магнитофоном ПЭВМ

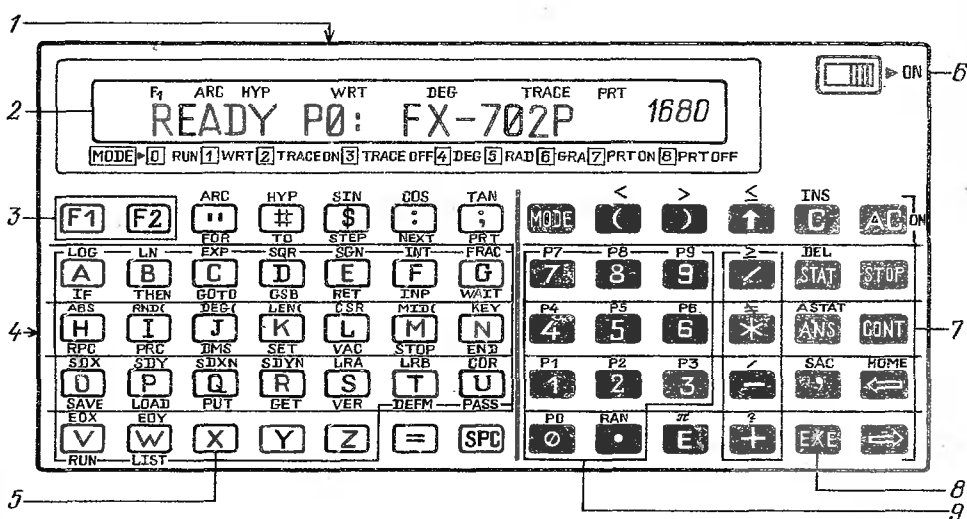


Рис. 1.3. Внешний вид ПЭВМ FX-702P (1 — регулятор контраста, 2 — дисплей, 3 — префиксные клавиши, 4 — разъем интерфейса, 5 — клавиши функциональные, 6 — выключатель питания, 7 — клавиша продолжения работы, 8 — клавиша ввода, 9 — цифровые клавиши)

FX-702P образует простейшую вычислительную микросистему (рис. 1.4), ориентированную на автоматизацию научных, инженерных, статистических и экономических расчетов умеренной сложности. Скорость счета ПЭВМ FX-702P в $25 \div 30$ раз превосходит скорость счета программируемых микрокалькуляторов Электроника-БЗ-34 [10].

§ 1.3. Персональные ЭВМ среднего класса (Home Computers)

ПЭВМ класса Home Computers обычно рассчитаны на подключение к бытовому кассетному магнитофону (накопителю) и телевизору (или дисплею). Они выполняются чаще всего в виде небольшого клавишного пульта с расположением клавиш, как у пишущей машинки. Такое решение обеспечивает резкое удешевление ПЭВМ (например, стоимость модели ПЭВМ ZX-81 английской фирмы Sinclair Radions Ltd. в 1983 г. составила 50 дол., что сделало ее дальнейшее производство нерентабельным).

Характеристики ряда простейших зарубежных ПЭВМ класса Home Computers даны в табл. 1.2. Эти ПЭВМ рассчитаны на работу с черно-белым телевизором и обеспечивают формирование алфавитно-цифровых знаков и графиков. Лишь одна из этих ПЭВМ — HP-75 содержит встроенный дисплей на одну строку и накопитель информации на магнитных картах.

Из отечественных ПЭВМ к данному классу относится бытовая персональная микроЭВМ Электроника-БК-0010 (бытовой компьютер). Эта ПЭВМ рассчитана на использование в качестве домашнего информационного центра, позволяющего проводить различные расчеты, создавать каталоги, справочники и словари, учебные программы, предоставлять пользователю игры и т. д.

ПЭВМ Электроника-БК-0010 выполнена на базе однокристального 16-разрядного микропроцессора K1801BM1 и имеет емкость ОЗУ 32 кбайт, в том числе ОЗУ пользователя 16 или 28 кбайт. Емкость ПЗУ 32 кбайт, а емкость ОЗУ экранной памяти 16 или 4 кбайт. ПЭВМ подключается к стандартному бытовому телевизору и кассетному магнитофону. На экран телевизора выводится 25 строк (одна служебная). При матрице знаков 8×8 в строке размещается 64 символа, а при матрице 16×8 — 32. Питание для ПЭВМ осуществляется от сети (220 В). ПЭВМ потребляет мощность не более 20 Вт. Ее размеры $370 \times 180 \times 70$ мм, масса не более 4 кг. Пульт ПЭВМ имеет 92 клавиши.

Графическая информация у ПЭВМ Электроника-БК-0010 отображается в формате 512×256 точек при двух градациях яркости и 256×256 точек при четырех градациях. Запись программ и данных на кассетный магнитофон происходит со скоростью 1200 бит/с. На стандартную кассету МК-60 можно записать информацию с объемом до 256 кбайт. Программируется ПЭВМ Электроника-БК-0010 на языках фокал (упрощенный язык, похожий на бейсик), бейсик и ассемблер (система команд, как у микроЭВМ Электроника-60). В первый год серийного производства (1985 г.) цена ПЭВМ Электроника-БК-0010 составила 540 р. [19].

Новое поколение ПЭВМ класса Home Computers (см. табл. 1.3) рассчитано на сопряжение с цветным дисплеем или телевизором. При этом ПЭВМ обеспечивают получение высококачественных многоцветных изо-

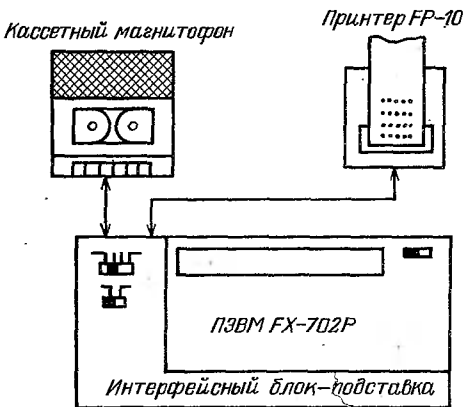


Рис. 1.4. Вычислительная микросистема на базе ПЭВМ FX-702P

МикроЭВМ FX-702P имеет удобные средства редактирования программ: замену и вставку любого символа, просмотр текста программ (листинга) построчно и в бегущем режиме, трассировку программ (режим TRACE) и др. Бейсик хранится в ПЗУ и вводится автоматически при включении машины. Дисплей отображает 20 знаков в формате матрицы каждого знака 5×7 , всего в строку можно включить 62 знака.

Характеристики ПЭВМ класса Pocket Computers непрерывно улучшаются. Некоторые модели таких ПЭВМ имеют вычислительные мощности, сравнимые с миниЭВМ 70-х годов.

Например, ПЭВМ HP-71B фирмы Hewlett Packard (см. табл. 1.1) имеет объем ПЗУ с внешними блоками до 320 кбайт, а ОЗУ — до 33,5 кбайт [17]. Эта микроЭВМ через интерфейс HP-IL может подключаться к 30 внешним устройствам. В их числе: телевизор или дисплей, отображающий одновременно 24 строки по 80 знакам, накопители информации на магнитной ленте и магнитных картах, принтер, акустические модемы, графопостроитель и др.

Большой объем ПЗУ (64 кбайт в минимальном варианте) обеспечивает реализацию расширенной версии бейсика, имеющей до 240 команд. Внешние ПЗУ могут содержать обширные библиотеки программ и банки данных пользователя, ориентированные на решение различных сложных профессиональных задач.

Для данного класса ПЭВМ характерен калькуляторный режим работы, обеспечивающий возможность проведения ряда расчетов без программирования, а также проверку исполнения отдельных команд.

Характеристики простейших персональных ЭВМ класса Home Computers

Характеристика	Тип ЭВМ			
	ZX-80	ZX-81	HP-75	PВ-700
Фирма и страна-изготовитель	Sinclair Radions Ltd. (Великобритания)	Sinclair Radions Ltd. (Великобритания)	Hewlett Packard (США)	Casio (Япония)
Разрядность микропроцессора	8	8	8	8
Емкость ПЗУ, кбайт	32	32	48+16	25
Емкость ОЗУ, кбайт	1	2	16+8	4+16
Число строк/знаков дисплея	Внешний *) 24/32	Внешний *) 24/32	1/32	4/20
Габариты, мм	299×178×51	299×178×51	—	200×85×20
Масса, кг	0,34	0,34	0,75	—
Периферийные устройства	ТД, П, НМЛ	ТД, П, НМЛ	НМЛ, АМ, П	НМЛ, П
Язык программирования	Бейсик	Бейсик	Бейсик	Бейсик

*) В качестве дисплея применяется телевизор.

Таблица 1.3

Характеристики некоторых зарубежных ПЭВМ класса Home Computers, сопрягаемых с цветным дисплеем (телевизором)

Характеристика	Тип ЭВМ					
	ZX-Spectrum	BBC-Micro A	VIC-20	ATARI-400	TI-99/4A	TRS-Colour
Емкость ОЗУ, кбайт стандартное/максимальное	16/48	16/32	5/29	16/16	16/48	16/32
Число цветов общее	8	8	16	16	16	9
Емкость ОЗУ, кбайт (для дисплея)	9	3	—	7	14	10
Число цветов, выводимых одновременно	8	4	16	4	16	8
Режим мерцания	Есть	Есть	Нет	Нет	Нет	Нет
Подстройка яркости	Есть	Нет	Нет	Есть	Нет	Нет
Установка графического режима пользователем	Есть	Есть	Есть	Нет	Есть	Нет
Полный набор кода ASC-II	Есть	Есть	Нет	Нет	Нет	Нет
Экран дисплея (строки/столбцы)	24×32	25×40	23×24	24×40	24×29	16×32
Автоповтор на всех клавишах	Есть	Есть	Нет	Есть	Есть	Нет
Графика повышенной разрешающей способности	Есть	Есть	Нет	Есть	Нет	Есть
Язык программирования	Бейсик, паскаль	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик

бражений, позволяя строить графики и рисунки с высокой разрешающей способностью, выводить различную текстовую и цифровую информацию в виде сложных многоцветных таблиц и т. д.

Одной из наиболее распространенных моделей таких ПЭВМ является микроЭВМ ZX-Spectrum английской фирмы Sinclair Radions Ltd. [11]. Клавишный пульт ZX-Spectrum (рис. 1.5), имеющий размеры 233×144×30 мм, обеспечивает ввод большинства

на одном гибком дискете 100 кбайт. Среднее время доступа 3—5 с.

У новой модели ПЭВМ ZX-Spectrum + число клавиш пульта увеличено до 58, что позволило реализовать ряд дополнительных текстовых и графических операторов и использовать новую расширенную версию языка бейсик. Емкость ОЗУ увеличена до 41,5 кбайт у основной модификации ПЭВМ. Накопитель на кассете с бесконечным рулоном ленты обеспечивает емкость внешней ПЗУ до

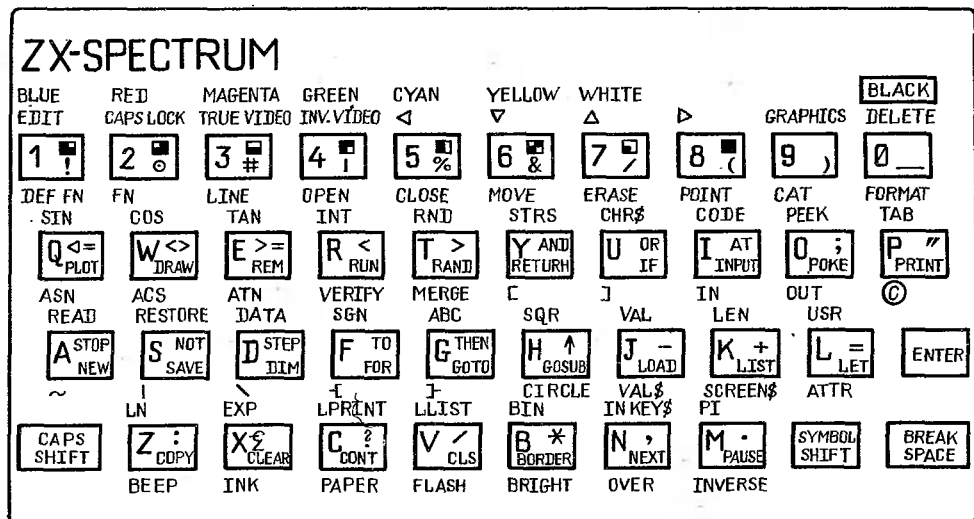


Рис. 1.5. Внешний вид ПЭВМ ZX-Spectrum класса Home Computers

команд расширенной версии бейсика нажатием одной-двух клавиш. Благодаря многократному использованию клавиш (с помощью префиксных клавиш) общее число их снижено до 40. Помимо стандартных команд бейсика пульт обеспечивает ввод до 37 графических символов и 22 кодов управления цветом. В пульт вмонтирован звуковой сигнализатор, дающий до 130 полутонов звука. Емкость ПЗУ 16 кбайт. При отображении графики в 8 цветах матрица экрана содержит 256×176 точек (при отображении алфавитно-цифровой информации формируется 24 строки с 32 символами в каждой). Клавишный пульт обеспечивает ввод полного набора символов по стандартному коду ASC-II.

Для ПЭВМ ZX-Spectrum разработаны специальные малогабаритные принтер и накопитель на диске. Принтер обеспечивает печать на рулоне бумаги шириной 10 см со скоростью 50 символов/с. По команде COPY принтер копирует сложные изображения с экрана телевизора (например, шахматную доску с фигурами).

Данная ПЭВМ может подключаться к кассетному магнитофону и обеспечивает скорость записи до 1500 бит/с. При этом данные или программы, занимающие объем ОЗУ 16 кбайт, записываются (и вводятся в ПЭВМ) за время 100 с. Миниатюрный дисковый накопитель обеспечивает скорость записи и считывания до 16 кбайт/с при объеме информации

85 кбайт на одну кассету. Загрузка программ емкостью 48 кбайт обеспечивается им за 9 с.

Как видно из табл. 1.3, различия у характеристик ПЭВМ класса Home Computers довольно несущественные. Все ЭВМ этого класса обеспечивают повышенные (в сравнении с микроЭВМ класса Pocket Computers) возможности в решении информационных задач, позволяющие использовать их в часы досуга (игры, просмотр компьютерных слайдов и т. д.). В то же время набор операторов для проведения математических и научных-технических вычислений у них ограничен — в лучшем случае он содержит операторы вычисления тригонометрических и обратных тригонометрических функций; натурального логарифма, экспоненциальной функции, квадратного корня и возведения в степень. В этом отношении некоторые типы ПЭВМ класса Pocket Computers (например, описанная выше модель FX-702P) значительно превосходят ПЭВМ класса Home Computers.

В последнее время появилось много моделей так называемых «портфельных» ЭВМ. Обычно они снабжены жидкокристаллическим многострочным индикатором, например 24 строки по 80 символов в каждой. Эти ПЭВМ занимают промежуточное положение между домашними и профессиональными ПЭВМ. В то же время ПЭВМ этого типа имеют большой объем ОЗУ (до 512 Кбайт) и развитые средства графики.

§ 1.4. Профессиональные ПЭВМ и вычислительные микросистемы индивидуального пользования

Профессиональные ПЭВМ рассчитаны на индивидуальное использование их профессиональными пользователями (математиками, программистами, учеными и инженерами, конструкторами, экономистами и т. д.). Обычно такие ПЭВМ предоставляются пользователям организациями и учреждениями, в которых они работают. Стоимость, размеры и масса профессиональных ПЭВМ могут заметно превышать таковые для ПЭВМ класса Pocket Computers и Home Computers. В то же время возможности этих машин должны быть довольно большими — лучшие типы профессиональных ПЭВМ не уступают по основным параметрам (объему памяти, скорости счета и номенклатуре периферийного оборудования) большим ЭВМ.

В табл. 1.4 приведены характеристики ряда зарубежных профессиональных ПЭВМ. Одной из первых моделей таких ПЭВМ следует считать модель HP-85 фирмы Hewlett Packard (США) [33]. В корпус средних размеров

заключены собственно ЭВМ, дисплей, принтер, таймер и кассетный накопитель. Все эти узлы потребляют от сети мощность менее 25 Вт, поэтому хороший тепловой режим работы ПЭВМ достигается без принудительной вентиляции, что полностью исключает шум при работе. Термопечатающее устройство этой ПЭВМ обеспечивает скорость печати 64 знака/с (две строки по 32 символа) с матрицей 5×7 .

Накопитель на магнитной ленте HP-85 имеет емкость памяти 210 кбайт при записи данных и 195 кбайт при записи программ, скорость записи и считывания 25,4 см/с, скорость поиска данных 152 см/с, время перемотки ленты 29 с, вероятность ошибки не более 1×10^8 . В НМЛ используются кассеты размерами $63,5 \times 82,5 \times 12,7$ см. В HP-85 встроены кварцевые часы, три программируемых таймера с временем выдержки от 1 до 9999999 с и звуковой сигнализатор. Дисплей этой ЭВМ может отображать одновременно алфавитно-цифровую и графическую информацию (в одном цвете). Предусмотрена память дисплея на 64 строки.

ПЭВМ HP-87 имеет емкость ОЗУ 16 кбайт, которая может наращиваться дополнительно

Таблица 1.4

Характеристики некоторых зарубежных профессиональных ПЭВМ и вычислительных микросистем

Характеристика	Тип ЭВМ					
	HP-85	TRS-80	Apple-II	PCjr	Professional	Formula-1
Страна-изготовитель	США	США	США	США	США	Япония
Емкость ОЗУ, мин/макс., кбайт	16/32	8/32	64/1000	64/128	256	64
Емкость ПЗУ мин/макс., кбайт	32/200	32	16	64	800	4/68
Число клавиш пульта	92	56	165 знаков	62	91	91
Размер ЭЛТ дисплея по диагонали, см	127	Дисплей жидкокристаллический	230 или 305	—	300 (цветной)	140
Матрица экрана при графике	256 × 192	240 × 64	—	640 × 200	1024 × 240	—
Число строк/столбцов	16/32	8/20	24/80	24/80	24/80 24/132	24/80
Встроенные устройства	Д, П, Т, НМЛ, ПЗУ	Д, Т, ПЗУ	НМД, ПЗУ	Т, ПЗУ, АМ, Т	Д, Т, ПЗУ	Д, П, Т, НМД (2 × 1 Мбайт, 2 × 0,7 Мбайт)
Периферийные устройства	ПЗУ	НМЛ, П	Д, П	Д, П, НМД	НМД	ПЗУ
Габариты, мм	158 × 419 × 452	220 × 300 × 50	380 × 397 × 157	—	610 × 375 × 165	424 × 255 × 212
Масса, кг	9,06	1,6	11	—	16	15
Основные языки программирования	Бейсик	Бейсик	Бейсик, паскаль, фортран и др.	Бейсик, паскаль, фортран и др.	Бейсик, фортран	Бейсик, фортран, паскаль и др.

на 32 кбайт. Емкость ПЗУ этой ПЭВМ достигает 0,5 Мбайт (в стандартном варианте 32 кбайт). В ЭВМ встроен дисплей на ЭЛТ с большим диаметром по диагонали (260 мм), обеспечивающий получение графических изображений с повышенной разрешающей способностью. Эта ЭВМ способна решать задачи линейного программирования с 200 переменными, системы линейных уравнений с порядком до 80 и т. д.

Портативная (в виде небольшого чемоданчика) ПЭВМ TRS-80 предназначена для руководителей и специалистов, часто бывающих в командировках. Дисплей у этой ПЭВМ жидкокристаллический (8 строк по 20 знаков). В ПЗУ TRS-80 зашито пять программ: интерпретатор расширенной версии бейсика (с графикой), обработка текстов, составление планов и расписаний, записная (адресная) книга и программа связи через телефонную сеть. ПЭВМ имеет ряд интерфейсов для связи с касетным магнитофоном, принтером и др.

Одной из наиболее распространенных ПЭВМ (их выпуск в США достиг 600 000 в год) является модель Apple-II (см. табл. 1.4) и ее модификация Apple-III фирмы Apple Computer Inc. В последнее время по объему выпуска с этими моделями конкурируют ПЭВМ IBM PC фирмы IBM (США). Эти ПЭВМ выполнены в виде малогабаритных вычислительных микросистем (процессор, клавишный пульт, дисплей, принтер и другие узлы). Модель PCjr имеет беспроводный клавишный пульт. Связь пульта с процессором осуществляется с помощью инфракрасного светового излучения. Модели PC фирмы IBM обладают программной совместимостью.

Большой емкостью ОЗУ (256 кбайт) и ПЗУ (до 5 Мбайт при расширении) отличается модель Professional (США). В нее встроен дисплей с необычно большой ЭЛТ (300 мм по диагонали), обеспечивающий получение многоцветных (4 095 градаций цвета) изображений с большой разрешающей способностью (матрица 1 024 × 240 элементов). Расширение ПЗУ достигается подключением накопителя на жестких (Винчестерских) дисках, имеющих емкость 5 Мбайт.

К моноблочной профессиональной ПЭВМ относится модель Formula-1 (Япония), ориентированная на проведение сложных вычислений. В эту ПЭВМ встроено два накопителя на гибких дисках (2 × 1 Мбайт и 2 × 0,7 Мбайт, запись двусторонняя), термопечатающее устройство и дисплей. Машина может выполнять функции программатора ЗУ, вставляемых в колодку на ее передней панели. Термопринтер обеспечивает печать со скоростью 0,8 строк/с (в одной строке 80 знаков). Машина имеет мощное программное обеспечение, в частности может программироваться на 25 языках высокого уровня, включая 5 расширенных версий бейсика, паскаля, фортран, кобол и др.

Принципиально новые средства общения с пользователем заложены в ПЭВМ Lisa и Macintosh фирмы Apple Computer Inc. (США). Эти ПЭВМ имеют два характерных признака: немерцающий цветной дисплей, который может работать в режиме долинкран-

(т.е. давать несколько не зависящих друг от друга изображений в специальных окнах), и устройство (контроллер) для перемещения маркера (курсора) дисплея типа «мышь». С последней особенностью связана принципиально иная процедура общения этих ПЭВМ с пользователем, чем у всех известных ранее ПЭВМ.

«Мышь» представляет собой небольшую коробочку, которую можно перемещать в любых направлениях по гладкой поверхности, например стола. В ней расположены датчики перемещений, с помощью которых можно формировать на экране дисплея сложные кривые либо указывать на отдельные фрагменты изображений. Клавишный пульт ПЭВМ Lisa и Macintosh заметен упрощен — в нем нет функциональных клавиш. Команды ЭВМ отображаются на дисплее наглядными изобразительными средствами (их совокупность образует «меню» команд). Для ввода любой команды («блода») достаточно подвести курсор к нужному рисунку и нажать специальную клавишу «мышь», вводящую команду в действие. Подобный подход к общению с ПЭВМ и ее программированию делает эти машины несовместимыми с ранее перечисленными моделями. Тем не менее он считается весьма перспективным, ибо облегчает диалог с пользователем, впервые работающим на ПЭВМ.

Отметим некоторые технические характеристики ПЭВМ Macintosh: емкость ОЗУ 128 кбайт, ПЗУ 64 кбайт, размер ЭЛТ дисплея 23 см по диагонали, число точек 512 × 342, встроены накопитель на гибких дисках малого диаметра ($\varnothing = 89$ мм фирмы Sony (Япония)) имеет емкость 0,4 Мбайт, интерфейс ПЭВМ обеспечивает передачу данных по шине Apple Bus со скоростью 192 кбайт/с, масса 9 кг, объем 20 дм³. ПЭВМ использует 32-разрядный микропроцессор 68 000 фирмы Motorola.

Из разработок отечественных ПЭВМ к данному классу можно отнести модель ПЭВМ Агат [14]. Ее характеристики: размер корпуса 500 × 350 × 180 мм, масса 9 кг, имеет откидной клавишный пульт с 59 клавишами. ПЭВМ Агат построена на 8-разрядном микропроцессоре, имеет ПЗУ емкостью 32 кбайт и ОЗУ емкостью 64, 128 и 256 кбайт. С помощью внешнего телевизора отображается 32 × 32 или 64 × 32 символов либо черно-белая графика с матрицей экрана 256 × 256 ячеек. При отображении восьмицветных изображений матрица составляет 128 × 128 точек, а шестнадцатичертовых — 64 × 64. Данные и программы могут записываться на обычный кассетный магнитофон (емкость 125 кбайт) или на НМД типа ЕС-5088 или ЕС-5089 (емкость 250 кбайт для двух НМД). Потребляемая ПЭВМ Агат мощность 60 Вт.

Некоторые модели отечественных микроЭВМ из-за относительно больших размеров, массы и стоимости можно условно отнести к ПЭВМ, поскольку они отвечают главному критерию таких вычислительных средств — представлению ресурсов в индивидуальном порядке. Подобные ЭВМ вместе с периферийным оборудованием образуют вычисли-

тельные микросистемы индивидуального пользования. В табл. 1.5 приведены краткие характеристики ряда отечественных вычислительных микросистем индивидуального пользования [27].

Массовое применение нашли вычислительные микросистемы на базе одноплатной микроЭВМ Электроника-С60, первые образцы которой программировались на упрощенной версии бейсика. В настоящее время для них используется расширенная версия бейсика дисковой операционной системы Рафос, в состав которой входят команды для работы и операций с символическими переменными. Эта версия бейсика совместима с применяемой в миниЭВМ серии СМ (бейсик-плюс [28]).

Для проведения научно-технических и других расчетов широко применяются системы подготовки программ (СПП) на базе специализированного вычислительного и управляющего устройства Электроника-Д3-28. Они имеют расширенную версию бейсика, причем расширение произведено в сторону увеличения числа операторов для вычисления математических функций. Все программы в справочнике даны на этой версии бейсика и могут легко переводиться на другие его версии.

В последние годы освоено массовое производство диалоговых вычислительных комплексов ДВК-1 и ДВК-2, построенных на

основе одноплатной микроЭВМ Электроника-НЦ-80-20 и дисплея 15 ИЭ-00-013-01 (клавиатура его показана на рис. 1.6). Этот дисплей применяется и в СПП на базе микроЭВМ Электроника-Д3-28. В комплекс ДВК-1 входят только упомянутые устройства. Комплексы ДВК-2 дополнительно укомплектованы накопителем на гибком диске ГМД-7012 и термопринтером.

Последние модели ДВК (ДВК-3 или Электроника-НЦ-80-20/3 [27]) имеют обширное периферийное оборудование. Емкость ОЗУ 64 кбайт, ПЗУ — 8 кбайт. Накопитель на гибком диске Электроника-6021 имеет емкость 440 кбайт (40 дорожек с 14 секторами, время перемещения головки с одной дорожки на другую 15 мс, время подвода головки 50 мс). ДВК-3 имеют быстродействие $500 \cdot 10^3$ операций/с, программируются на языках бейсик, фортран, паскаль. Ассемблер этих микроЭВМ имеет систему команд, принятую для микроЭВМ Электроника-С60. В комплексы входят принтеры (DZM-180, Д-100, PRT-80 или СМ-6310), графопроектор и другие периферийные устройства.

В значительной мере требованиям к профессиональным ПЭВМ отвечают микроЭВМ серийного производства Искра-226. ЦП этой ЭВМ с быстродействием $2 \cdot 10^6$ микрокоманд/с и разрядностью 8/16 имеет обращение к ОЗУ

Таблица 1.5

Характеристики некоторых отечественных вычислительных микросистем индивидуального пользования

Характеристика	Тип системы или микроЭВМ				
	Электроника				Искра-226
	ТЗ-29	ТЗ-29М	ДЗ-28	НМС *) 011 00.1	
Общая емкость ОЗУ, кбайт	64	288	16—128	56	112—176
Емкость ОЗУ пользователя, кбайт	32	256	—	48	64—128
Хранение программы-интерпретатора	В ПЗУ	В ПЗУ	На магнитной ленте	В ПЗУ	В ПЗУ и на диске
Среднее быстродействие процессора, тыс. оп/с	100	600	—	400	800
Общая масса оборудования, кг	117	120	85	116	166
Общая потребляемая мощность, Вт	700	1000	500	800	2000
Язык программирования	Бейсик	Бейсик	Бейсик	Бейсик	Бейсик, паскаль
Встроенные устройства	НМЛ	НМЛ	НМЛ	Д	Д
Периферийные устройства	Д, ТП, П	Д, ТП, П	Д, ТП, П, НМД, ГП	ТП, НМД, ГП, ГД	П, НМД, НМЛ, ГД

*) В тексте диалоговый вычислительный комплекс Электроника НМС 01100.1 на основе дисплея 15 ИЭ-00-013-01 и одноплатной микроЭВМ Электроника НМС 11100.1 обозначен как ДВК. Масса ДВК без периферийных устройств 36 кг, потребляемая мощность 275 Вт.

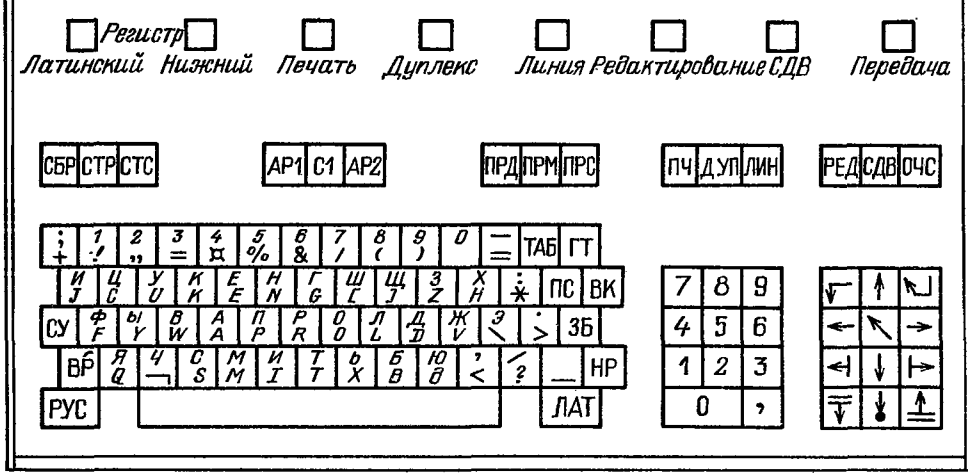


Рис. 1.6. Клавишный пульт дисплея 15 ИЭ-00-013-01, применяемого в диалоговых вычислительных комплексах ДВК-1, ДВК-2 и в системах подготовки программ на базе микроЭВМ Электроника-Д3-28

емкостью 64 кбайт и ПЗУ 48 кбайт (основной вариант). Встроенный алфавитно-цифровой и графический дисплей на ЭЛТ с размером 31 см по диагонали отображает 24 строки по 80 символов или 256 × 560 точек графиков. К нему может подключаться внешнее видеоконтролирующее устройство ВК40В60. Кассетный накопитель Искра-005-33 имеет емкость 80 кбайт, скорость обмена 200 байт/с и среднее время доступа 180 с. Внешние накопители (НМД и НМЛ) увеличивают емкость ОЗУ на 5 и 10 Мбайт. ЭВМ может комплектоваться графопостроителем и принтером со скоростью печати до 180 знак/с. Вычислительные микросистемы на базе микроЭВМ Искра-226 выпускаются в нескольких модификациях.

Приведенные в табл. 1.5 вычислительные микросистемы не обеспечивают возможности получения цветных графических изображений. В настоящее время завершена разработка профессиональной отечественной ПЭВМ, аналогичной описанной выше модели Professional (США) и обеспечивающей получение цветных графических изображений с высокой разрешающей способностью.

Профессиональные ПЭВМ и вычислительные микросистемы могут программироваться на многих языках: бейсик, фортран, паскаль, алгол и др. Однако наиболее распространен бейсик, обеспечивающий возможность работы на этих системах пользователей, не являющихся специалистами по программированию.

§ 1.5. Периферийное оборудование персональных ЭВМ.

Ниже даны краткие сведения о типовом периферийном оборудовании ПЭВМ, рассчитанные на лиц, занятых их эксплуатацией.

Одним из важнейших узлов ПЭВМ является дисплей — устройство на основе электронно-лучевой трубки, обеспечивающее отображе-

ние алфавитно-цифровой и графической информации. В качестве дисплея ПЭВМ класса Home Computers обычно используются бытовые телевизоры. В состав ПЭВМ входит адаптер, преобразующий сигналы ПЭВМ в высокочастотные сигналы, принимаемые телевизором по одному из телевизионных каналов. Весьма нежелательно совпадение номера этого канала с номером канала местной телевизионной станции, поскольку при этом возможны взаимные помехи — ПЭВМ может создавать помехи другим телевизорам, а местная станция — искажать прием сигналов с ПЭВМ. Качество изображения можно значительно повысить, подключив выход ПЭВМ ко входу видеосигналов телевизора (т. е. минуя преобразование видеосигнала в высокочастотный сигнал).

Сложные ПЭВМ могут комплектоваться специальными дисплеями и мониторами, обеспечивающими высокое качество изображения (монитором обычно называют дисплей с клавишным пультом).

Другим распространенным устройством для подключения к ПЭВМ является кассетный магнитофон. Хотя многие ПЭВМ не критичны к качеству магнитофона, целесообразно применять магнитофоны класса не ниже третьего. Магнитофоны более высокого класса и специальные кассетные накопители информации обычно имеют электромагниты управления лентопротяжным механизмом. Они обеспечивают возможность автоматического включения и выключения магнитофона в ходе ввода программ в ПЭВМ и их вывода на магнитофон. Для подключения магнитофона к ПЭВМ используется специальный адаптер, содержащий интерфейс сопряжения (иногда этот интерфейс входит в ПЭВМ). Основным недостатком кассетного магнитофона — значительное время поиска нужной информации, что связано с медленной протяжкой магнитной ленты.

К массовым периферийным устройствам профессиональных ПЭВМ относятся накопители информации на вращающихся магнитных дисках. Они бывают двух типов: со сменными гибкими дисками и жесткими (Винчестерскими). В основном в ПЭВМ применяются диски малого диаметра ($\varnothing = 133$ мм). Запись блоком магнитных головок производится на одну или обе стороны диска, покрытого ферромагнитным материалом. На каждой стороне диска имеется несколько десятков дорожек записи, каждая из которых разбита на секторы. При записи и считывании информации пользователь задает в виде специальных кодов сторону диска, номер дорожки и номер сектора, переключение дорожек производится быстро. Емкость одного диска лежит в пределах от 0,2 до $50 \div 100$ Мбайт (большие цифры — для жестких дисков с диаметром 203 мм). Скорость обмена данными весьма высока — до $0,125 \div 10$ Мбайт/с, что является большим достоинством НМД по сравнению с накопителями на магнитной ленте. В последнее время наблюдается тенденция перехода к малогабаритным НМД с диаметрами дисков 76, 82,5, 85 мм и меньше.

Помимо накопителей информации на стандартных кассетах с магнитной лентой и магнитных дисках в некоторых ПЭВМ применяются специальные типы накопителей: на микрокассетах (уменьшенный вариант обычной кассеты), кассетах типа Cartridge с бесконечным рулоном ленты, свернутой в кольцо, и на магнитных картах. Однако широкого распространения такие накопители не получили. В перспективе возможно оснащение ПЭВМ накопителями информации на оптических дисках с лазерной записью и считыванием.

Из другого периферийного оборудования следует отметить принтеры — устройства для печати алфавитно-цифровой, а иногда и графической информации. Принтеры подключаются к ПЭВМ через интерфейс (обычно с передачей информации в параллельном

коде). Скорость печати принтеров для ПЭВМ составляет от нескольких десятков до нескольких сотен знаков в секунду. Наиболее распространены мозаичные принтеры, обеспечивающие точечную печать знаков с матрицей 5×7 элементов. Однако качество такой печати не считается достаточно высоким. Поэтому находят применение и принтеры, работающие по типу пишущей машинки. Высокое качество печати в сочетании с высокой скоростью имеют принтеры с шаровой печатающей головкой. У них знаки распределены по поверхности шара и нужный знак выбирается поворотом печатающей головки. Принтеры, разом печатающие целую строку, применяются в основном в больших ЭВМ.

Для высококачественного документирования графической информации применяются графопостроители. Имеются графопостроители с числом перьев до $6-12$, обеспечивающие построение многоцветных графиков с высокой разрешающей способностью (до 0,1 мм и выше) и большой точностью задания координат (погрешность до $0,2 \div 1\%$). Скорость построения графиков лежит в пределах от 2 до 50 см/с.

Преобразование координат точки (или точек графика) в цифровой код обеспечивают специальные планшеты. С этой целью применяются и устройства типа «мышь», содержащие поворотные датчики, реагирующие на перемещение «мыши» по гладкой поверхности. Погрешность преобразования для таких устройств составляет обычно доли процента.

Для связи ПЭВМ с внешними накопителями информации через телефонную сеть служат акустические модемы: специальные микрофоны и излучатели, работающие на высоких звуковых и ультразвуковых частотах. Акустические модемы могут применяться в виде отдельных блоков или встраиваться в корпус ПЭВМ. Тогда в акустическом модеме закрепляется трубка телефонного аппарата, что обеспечивает акустический контакт с ПЭВМ.

§ 2.1. Алфавит и основные операторы языка бейсик

Назначение языка бейсик (BASIC). Наименование BASIC является сокращением английских слов «Beginner All-purpose Symbolic Instruction Code», что в переводе означает «многоцелевой язык символических команд для начинающих». BASIC был задуман разработчиками как язык, доступный для усвоения студентами гуманитарных вузов. Однако простота грамматики и синтаксиса, его схожесть с фортраном привели к широкому распространению бейсика в практике математических и научно-технических расчетов.

Бейсик — язык интерпретирующий. Работа на бейсике организуется с помощью специальной программы-интерпретатора, которая записана в ПЗУ или загружается в ОЗУ с внешнего накопителя — магнитной ленты или диска. Эта программа переводит каждый оператор бейсик-программы на машинно-ориентированный язык конкретной ЭВМ.

Бейсик — язык высокого уровня. Многие версии бейсика имеют принципиальные отличия в основных операторах. Поэтому возможен простой перевод программ с одной версии бейсика на другую. Однако не следует преуменьшать и различия между разными версиями. Так, простые версии бейсика имеют всего 30—40 операторов, тогда как у сложных их число достигает нескольких сотен. При переводе программ следует учитывать различия в записи отдельных операторов, в их наборе и правилах синтаксиса. Различия в версиях бейсика усугублялись использованием его для программирования наиболее массовых видов ЭВМ — ПЭВМ, число типов которых составляет многие тысячи.

Бейсик — язык диалоговый. Это означает, что программными средствами на бейсике можно организовать диалоговый режим работы ПЭВМ с пользователем. При этом ПЭВМ сообщает пользователю о том, какими возможностями она обладает, как ею пользоваться, в каком порядке должна выполняться данная конкретная программа и т. д. Диалог задается пользователем и в его программах.

Алфавит бейсика содержит 26 прописных латинских букв, цифры от 0 до 9, точку, запятую, кавычки, знаки арифметических операций (+, —, /, * (умножение) и \uparrow (возве-

дение в степень)), знаки отношения двух величин (>, <, =), скобки и другие специальные знаки. Версии бейсика для отечественных микроЭВМ имеют в алфавите и русские буквы.

Переменные бейсика делятся на 4 типа.

1. Простые переменные, обозначаемые латинскими буквами, например A, B, C, ..., X, Y, Z.

2. Простые переменные, обозначаемые латинскими буквами с одной цифрой после них, например A1, D2, ..., Z9.

3. Индексированные переменные одномерных массивов вида A(α), B(α), ..., Z(α), где α — число, простая переменная или арифметическое выражение.

4. Индексированные переменные двумерных массивов вида A(α , β), B(α , β), ..., Z(α , β), где α и β — числа, простые переменные или арифметические выражения.

Максимальные значения α и β зависят от объема ОЗУ (обычно 256).

Примеры. X(8) — переменная одномерного массива X с номером 8, Y(2,3) — переменная двумерного массива, стоящая в третьей строке и четвертом столбце (нумерация строк и столбцов начинается с 0), Z(1, J*K) — переменная двумерного массива, номер строки которой задается значением целой части переменной I, а номер столбца — значением целой части произведения J*K.

Числа могут вводиться в программу в качестве вещественных констант либо вводиться с пульта для задания числовых значений переменных. Числа задаются в естественном виде с разделительной точкой, а не запятой. Ноль перечеркивается наклонной чертой, что позволяет отличить его от буквы O. Порядок числа указывается после буквы E, например число 1,234 можно ввести в виде 1.234; .1234E1; 123.4E-2 и т. д. Перед мантисой или порядком могут стоять знаки + и —, обычно знак + опускается. Например, число -0,000097 записывается в виде -0.000097; -.000097; -97E-6; -9.7E-05 и т. д. Первый ноль мантисы обычно можно не вводить. Максимальный порядок чисел зависит от типа ЭВМ и лежит в пределах от 30 до 252.

Операторы бейсика — допустимые предписания, предназначенные для задания ЭВМ операций обработки информации. Операторы записываются в виде слов или предложений

большими латинскими буквами без пробелов в словах (у некоторых ПЭВМ пробелы игнорируются). Существуют также служебные операторы (директивы), исполняемые сразу после их ввода с пульта (некоторые из них могут выполняться и по программе).

Программа на бейсике есть последовательность операторов, записанных по определенным правилам и обеспечивающих решение заданной задачи на ЭВМ. В бейсике операторы записываются построчно. В одной строке может быть один оператор или несколько, разделенных специальным знаком (двоеточием : или наклонной чертой \). Строки должны иметь последовательную нумерацию. Рекомендуется задавать номера строк с пропусками, например 5, 10, 15, 20 и т. д. или 10, 20, 30 и т. д. Это позволит при необходимости вставить между ними новые строки. Максимальный номер строки зависит от типа ПЭВМ.

Таблица 2.1

Символы арифметических операций бейсика

Символ	Пример	Операция.	Приоритет операции
↑	A ↑ B	Возведение в степень	Первый
*	A * B	Умножение	Второй
/	A / B	Деление	Второй
+	A + B	Сложение	Третий
-	A - B	Вычитание	Третий

Символы арифметических операций бейсика даны в табл. 2.1. Там же отмечен приоритет операций, т. е. порядок их выполнения при вычислении арифметических выражений. Приоритет может быть изменен вводом скобок.

Операторы задания массивов DIM и COM.

Одномерные и двумерные массивы занимают в ОЗУ ЭВМ, определенную область памяти, обозначаемую как DIM или COM. Чтобы в эту область не попали другие данные, перед вводом и использованием массивов нужно зарезервировать для них место. Это делается с помощью операторов DIM и COM, после кото-

рых указывается переменная и скобки с указанием размеров массива. Так, полный оператор DIM A (9) резервирует в памяти место для одномерного массива A из 10 переменных от A(0) до A(9). Оператор DIM C(5, 6) резервирует место под двумерный массив C, содержащий шесть строк и семь столбцов (включая нулевые).

Для обозначения размерности массивов можно использовать арифметические выражения (причем их дробная часть отбрасывается), а также числовые значения переменных. Например, оператор DIM R (I * N, K) резервирует область памяти DIM под двумерный массив R с максимальным номером строки, равным целой части произведения I * N и максимальным номером столбца, равным целой части числового значения переменной K.

Операторы выполнения математических функций на бейсике приведены в табл. 2.2. Аргумент функции должен заключаться в скобки. Им могут быть числа, переменные и арифметические выражения, например:

SIN(30) — синус угла в 30 градусов,
 SIN(Z) — синус угла, заданного аргументом в виде числового значения переменной Z,
 EXP(SQR(2)) — значение числа $e^{\sqrt{2}} = \exp \sqrt{2}$.

Аргумент и значение функций не должны выходить за пределы определения функции и за пределы разрядной сетки ЭВМ, иначе будет индифферентна ошибка. Вычисление функции имеет приоритет перед арифметическими операциями.

Исполняемые по программе основные операторы версии бейсика для микроЭВМ Электроника-60, Электроника-ДЗ-28 даны в табл. 2.3. Рассмотрим их более детально.

Оператор LET (от слова letter — буква) служит для присвоения буквенным переменным, указанным слева от знака равенства, числового значения константы, переменной или арифметического выражения, записанных справа от знака равенства.

Оператор DATA (от слова data — данные) служит для ввода в память ЭВМ определенной последовательности чисел. Они даются списком с разделительными запятыми.

Таблица 2.2

Операторы вычисления математических функций версии бейсика микроЭВМ Электроника-60

Запись оператора	Запись функции	Наименование функции	Запись оператора	Запись функции	Наименование функции
SIN(X)	$\sin x$	Синус X	RND (x)	rnd x	Генерация случайных чисел со значением от 0 до 1
COS(X)	$\cos x$	Косинус X	SGN(X)	sign x	Сигнум-функция (+1 при $x > 0$, 0 при $x = 0$ и -1 при $x < 0$)
ATN(X)	$\arctg x$	Арктангенс X	FN α(X)	—	Функция, заданная пользователем
SQR(X)	\sqrt{x}	Корень квадратный из X	EXF(X)	—	Внешняя функция
EXP(X)	$e^x = \exp x$	Экспонента X			
LOG(X)	$\ln x$	Натуральный логарифм X			
ABS(X)	abs x	Абсолютное значение X			
INT(X)	int x	Целая часть X			

Основные операторы языка бейсик

Операторы	Примеры записи	Пояснения
LET	LET A=2 LET X=Y LET Z= α	Переменной A присваивается значение 2 Переменной X присваивается значение переменной Y Переменной Z присваивается значение арифметического выражения α
DIM COM	DIM A(5) COM A(I,J) DIM A(α ,L)	Задание одномерного массива A из шести переменных A(0), A(1), ..., A(5) Задание двумерного массива с максимальным номером строки I и столбца J Задание двумерного массива с максимальным номером строки, равным целой части арифметического выражения α , и столбца с максимальным номером, определяемым численным значением переменной L
DATA	DATA 1, 2, 3, 4	Ввод чисел 1, 2, 3, 4
READ	READ A, B, C, D	Присвоение переменным A, B, C и D численных значений, заданных оператором DATA
RESTORE	RESTORE	Поворот стека данных
INPUT	INPUT A INPUT 'A='A INPUT A, B, C	Ввод значения A с пульта после появления знака ? Ввод значения A после печати пояснения A= Ввод значений A, B и C, разделенных запятыми
PRINT	PRINT 2 PRINT A, B, C PRINT 'A='A PRINT 'F=' α	Печать числа 2 Печать численных значений переменных A, B и C Печать численного значения A после комментария A= Печать численного значения арифметического выражения α после комментария F=
GOTO	GOTO n	Безусловный переход к строке с номером n
IF ... THEN IF ... GOTO	IF α r β THEN n IF α r β GOTO n IF α r β THEN γ	Условный переход к строке n, если справедливо отношение r (табл. 2.4) Выполнение операций γ , если справедливо отношение r
FOR ... TO ... STEP ... NEXT	FOR α = β TO γ STEP δ NEXT α FOR α = β TO γ NEXT α	Организация цикла с управляющей переменной α , меняющейся от начального значения β до конечного γ с шагом δ (β , γ и δ — числа, значения переменных или арифметических выражений) Организация цикла с шагом $\delta = +1$
GOSUB	GOSUB n	Обращение к подпрограмме, начинающейся со строки n
RETURN	RETURN	Возврат из подпрограммы

Операторы	Примеры записи	Пояснения
REM (REMARK)	REM 'КОММЕНТАРИЙ'	Печать комментария при выводе листинга программы
STOP	STOP	Остановка вычислений в любом месте программы
END	END	Конец программы и останов в конце программы
DEF FN(V) (параметр) = α	DEF FN Y(X) = SIN (X)/X	Задание функции α пользователя; в примере $Y(X) = \sin x/x$

Оператор READ (read — читать) служит для чтения чисел из блока данных DATA и присвоения переменным своего списка^b численных значений.

Пример. В следующей программе численные значения переменных будут такими: $A=1$; $B=2$; $C=3$; $D=4$ и $E2=-5 \cdot 10^{-8}$.

```
10 DATA 1,2,3,4,-5E-8
20 READ A,B,C,D
30 READ E2
```

Оператор RESTORE (от слова restore — возвращать) служит для поворота стека данных, который формируется в ОЗУ ЭВМ оператором DATA (стек можно уподобить обойме револьвера, если числа отождествлять с патронами). Этот оператор необходим, если, например, одно из чисел в списке DATA нужно присвоить разным переменным.

```
10 DATA 1,2,3,4,-5E-8
20 READ A,B,C,D,E2
30 RESTORE
40 READ F,L,L
```

В приведенной программе вначале переменным стека оператора READ в строке 20 присваиваются отмеченные выше численные значения. Затем оператор RESTORE в строке 30 обеспечивает поворот (возврат) стека. В результате переменной F присваивается значение 1, а переменной L — сначала значение 2, а затем 3.

Оператор INPUT (input — ввод) служит для ввода чисел с пульта по запросу ЭВМ. Он сопровождается печатью знака вопроса (?) или комментария, заключенного в апострофы (или кавычки), после слова INPUT, а также для присвоения числовых значений списку переменных.

Примеры.

1) 10 INPUT A

Печатается знак вопроса, и ЭВМ ждет число, значение которого (после ввода) будет присвоено переменной A.

2) 20 INPUT A, B, C, D

Печатается знак вопроса, после чего надо ввести четыре числа, разделяя их запятыми.

3) 30 INPUT ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N='N

Печатается запрос ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N=, после которого надо ввести значение N.

4) 10 INPUT ВВЕДИТЕ A, B, A, B

Печатается указание ВВЕДИТЕ A, B, после чего следует ввести значения A и B, разделив их запятой.

У некоторых микроЭВМ, например Электроника-60, встречаются версии бейсика, у которых применение комментария после оператора INPUT не предусмотрено. В этом случае для вывода комментария используется оператор PRINT (см. ниже).

Оператор PRINT (от слова print — печать) служит для задания вида печати (специальным знаком после него), печати комментариев, заключенных в апострофы (или кавычки), печати числовых значений констант, переменных и арифметических выражений, входящих в список этого оператора.

Примеры.

1) 50 PRINT 125, 98, 36

Печатаются числа 125 98 36.

2) 50 PRINT A, B, C

Печатаются числовые значения переменных A, B, C. Так, если $A=125$, $B=98$ и $C=36$, то печать имеет вид, аналогичный приведенному выше.

3) 50 PRINT ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ'

Печатается комментарий ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

4) 100 PRINT 'A='A2

Печатаются комментарии и значение переменной A2. Если $A2=125$, печатается $A=125$.

5) 60 PRINT 'F='SQR (A*B+C)

Печатается $F=\alpha$, где α — числовое значение арифметического выражения $\sqrt{AB+C}$.

6) 70 PRINT

Печатается пустая строка (т. е. пробелы).

Если в списке оператора PRINT в качестве разделителя используется запятая, то каждое число (см. примеры 1) и 2)) печатается в соответствующей зоне одной строки. Послед-

няя разделяется обычно на пять зон печати. Если чисел больше пяти, то последующие печатаются с новой строки, и т. д. Для более плотной печати в качестве разделителя можно использовать знак ; (точка с запятой). Тогда очередное число будет следовать за предыдущим с минимальным пробелом.

Тип печатающего устройства указывается специальными знаками после оператора PRINT. Например, для системы подготовки программ на базе микроЭВМ Электроника-Д3-28 оператор PRINT после включения системы обеспечивает вывод комментариев и чисел только на дисплей. Для печати принтером после оператора PRINT следует поставить знак #1, а для обратного переключения индикации на дисплей — знак #0.

Пример.

```
10 PRINT#0'ВЫЧИСЛЕНИЕ ЭКСПОНЕНТЫ'
20 INPUT X
30 PRINT#1'X='X,'EXP(X)='EXP(X)
40 END
```

В этом примере на экран дисплея будет выведено сообщение ВЫЧИСЛЕНИЕ ЭКСПОНЕНТЫ, а значения X и EXP(X) будут напечатаны принтером. Таким образом, первоначально введенный знак #0 или #1 сохраняется до замены его другим, повторять этот знак в последующих операторах PRINT не нужно.

Формат печати чисел также может указываться специальным знаком в составе оператора PRINT. Так, для упомянутой выше системы возможны следующие форматы:

!F n_1 . n_2 ! — выводится знак числа, n_1 цифр до десятичной точки, десятичная точка, n_2 цифр после десятичной точки, символ порядка E, знак порядка и две значащие цифры порядка;

n_1 . n_2 ! — выводится знак числа, n_1 цифр до десятичной точки, десятичная точка и n_2 цифр после нее (если число не умещается в формате, выводится столько символов, сколько позиций отведено на печать);

!E! — число выводится в форме с плавающей запятой и содержит знак числа, точку, 12 цифр мантиссы, символ порядка E, знак порядка и две цифры порядка.

При начальном включении системы устанавливается формат IF 1.91. При смене формата новый формат запоминается и действует до замены его другим.

Внимание! Синтаксис и вид операторов INPUT и PRINT у разных моделей ПЭВМ могут иметь некоторые различия. Например, комментарии могут вводиться в кавычках или апострофах, различны символы установки формата и типа печати. Поэтому общее знакомство с этими операторами следует завершить уточнением формы их записи для каждой конкретной ПЭВМ.

Оператор GOTO n (от слов go to — иди к) обеспечивает безусловный переход к строке с номером n . В программе

```
10 INPUT'X='X
20 LETY=SIN(X)
25 PRINT'Y='Y
30 GOTO 10
```

после ввода X и вычисления $Y = \sin X$ оператор GOTO 10 в строке 30 обеспечивает переход к строке 10, т. е. к вводу нового значения X по оператору INPUT. Таким образом, программа зацикливается.

Операторы IF и THEN (if — если, then — тогда) служат для организации условных переходов. Общая форма их записи (вместо слова THEN возможно GOTO).

```
IF  $\alpha$   $r$   $\beta$  THEN  $n$  (или  $\gamma$ )
IF  $\alpha$   $r$   $\beta$  GOTO  $n$ 
```

где α , r и γ — арифметические выражения, n — номер строки, r — символ отношения α и β (условия перехода) (табл. 2.4). Вместо α и β могут стоять переменные и числа. Если предписанное отношение выполняется, происходит

Таблица 2.4

Символы отношения r для операторов IF...THEN

Символ бейсика	Математический символ	Пример записи оператора	Значение символа
=	=	A=B	Равно
<=	<=	A<=B	Меньше
<<=	<=	A<<=B	Меньше или равно
>	>	A>B	Больше
=>	>=	A=>B	Больше или равно
<>	≠	A<>B	Не равно

переход к строке с адресом n (или выполняют арифметические операции в выражении γ). В противном случае выполняется последующая за операторами IF... THEN часть программы.

Пример.

```
10 INPUT'X='X
20 IF X=0 THEN 50
30 LETF=SIN(X)/X
40 PRINT'F='F:GOTO 10
50 LETF=1:GOTO 40
```

В приведенной программе вычисляется функция $F = \sin x/x$. При $x=0$ получаем неопределенность вида 0/0, а поскольку деление на 0 машина не производит, нужно этот случай исключить. Для этого в строке 20 значение x сравнивается с нулем. Если $x=0$, происходит переход к строке 50, где переменной F присваивается известное значение 1.

Операторы организации циклов FOR...TO и STEP образуют цикл по схеме

```
FOR  $\alpha$ = $\beta$  TO  $\gamma$  STEP  $\delta$ 
(тело цикла)
```

```
NEXT  $\alpha$ 
```

Эта запись означает следующее: для (for) переменной α , имеющей начальное значение β и меняющейся до (to) конечного значения γ с шагом (step) δ обеспечивается выполнение цикла, завершаемого оператором NEXT α (следующий). Если α достигает значения γ ,

присходит выполнение следующего за циклом оператора. Переменная α должна быть простой, а β , γ и δ могут быть числами, переменными или значениями арифметических выражений. Отсутствие в цикле оператора STEP автоматически придает величине δ значение $\delta = +1$. Примеры применения операторов организации циклов даны в § 2.4.

Оператор обращения к подпрограмме GOSUB n (от слов go subroutine — идти к подпрограмме) обеспечивает переход к выполнению подпрограммы, начинающейся со строки n . Подпрограммы применяются в том случае, когда нужно неоднократно выполнить одну и ту же группу операторов.

Оператор возврата из подпрограммы RETURN (return — возвращение) обеспечивает возврат из подпрограммы после ее выполнения к оператору, следующему за обращением GOSUB n . Оператор RETURN ставится в конце подпрограммы.

Циклы и подпрограммы могут быть вложенными друг в друга (более подробно см. § 2.4).

Оператор программных комментариев REMARK (или сокращенно REM, от слова remark — комментировать). Этот оператор служит для ввода в программу пояснений — комментариев, заключенных в апострофы или кавычки (или без них). Эти комментарии выводятся на печать только при выводе листинга программы.

Применение комментариев делает сложные программы более наглядными.

Оператор останова STOP используется для остановки вычислений в любом месте программы. Обычно при этом ПЭВМ выдает комментарий с указанием, в какой строке произошел останов, например:

```
ОСТАНОВ В СТРОКЕ 100
ГОТОВ
```

Оператор окончания программы END служит для фиксации конца программы, что необходимо, например, при записи программы на магнитную ленту или диск. Если этот оператор поместить в середине программы, будет записана только ее часть — до оператора END. Оператор END также обеспечивает останов вычислений.

Оператор задания функций пользователем — DEF FN (переменная) (параметр) = α , где α — собственно функция. Например, выражение DEF EN $Y(X) = \text{SIN}(X)/X$ задает функцию $Y(X) = \text{SIN}(X)/X$, которую можно использовать как обычную стандартную функцию.

Служебные операторы бейсика приведены в табл. 2.5. Вместо оператора стирания DELETE (от слова delete — уничтожать) часто используется оператор CLEAR (clear — стирать). Оператор SAVE записи программы на носитель и операторы загрузки программы в ЭВМ с носителя (OLD или LOAD) часто дополняются указаниями на тип носителя (перфоленга, магнитофонная кассета, магнитный диск и т. д.) и на место записи или считывания (номер дорожки, сектора и т. д.). Полный вид этих операторов у различных ПЭВМ может иметь некоторые принципиальные различия. Приведенные ниже примеры относятся к работе с системой подготовки программ на базе микроЭВМ Электроника-ДЗ-28.

При записи текста программы на магнитную ленту интерпретатор формирует так называемый программный файл, состоящий из заголовка, записи строк программы и закрывающей файл записи. Для записи программы используется оператор SAVE (хранить).

Примеры.

SAVE 50, 150 — записываются строки от 50 до 150 безымянной программы;

SAVE 'СТАТИСТИКА' — записывается вся программа с именем СТАТИСТИКА;

SAVE — записываются все строки (от 1 до 7999) программ, имеющихся в ОЗУ;

SAVE END — служебная запись, означающая конец ленты.

Чтение программных файлов с магнитной ленты и их загрузка в ОЗУ производится оператором LOAD (загрузить).

Примеры.

LOAD 'α' HC1, HC2 — в ОЗУ загружаются строки программы с именем α от HC1 до HC2;

LOAD 'СТАТИСТИКА' — загружается вся программа с именем СТАТИСТИКА.

Следует отметить, что имя программ распознается только по шести начальным символам, включая пробелы. Если имя файла

Таблица 2.5

Служебные операторы бейсика

Запись оператора	Действие оператора
DELETE n	Стирание строки с номером n
DELETE m, n	Стирание строк с номерами от m до n
LIST	Вывод листинга (текста) всей программы
LIST n	Вывод листинга строки с номером n
LIST m, n	Вывод листинга части программы (строки от m до n)
OLD или LOAD	Загрузка программы с носителя
RUN	Пуск программы (вычислений)
SAVE	Запись программы на носитель
СКЛ/Р	Остановка программы с печатью знака \uparrow
СКЛ/У	Стирание вводимой строки
ЗБ	Забой (стирание) одного символа

не задано, при выполнении оператора SAVE произойдет загрузка очередного файла.

Для записи на магнитную ленту блока данных применяется оператор DATA SAVE с последующим списком блока. Например, оператор DATA SAVE B(3), SIN (X+Y) * Z, C обеспечивает запись на ленте значений переменной B(3) одномерного массива B, выражения Z * SIN (X+Y), переменной C.

Блоки могут объединяться в файлы данных, которые начинаются с оператора (заголовка) DATA SAVE OPEN и заканчиваются оператором DATA SAVE END. После заголовка может указываться имя файла, заключенное в апострофы.

Чтение с магнитной ленты блока данных с присвоением их переменным производится оператором DATA LOAD со списком переменных, разделенных запятыми. Оператор DATA LOAD 'α' обеспечивает поиск на магнитной ленте файла с именем α. Для пропуска некоторого числа блоков или файлов используется оператор SKIP (перепрыгнуть), после которого указывается это число, например, оператор SKIP 2 обеспечивает пропуск двух блоков данных, а оператор SKIP 3 — пропуск трех файлов данных. Перемотка ленты в начало производится оператором REWIND (перемотать).

Операторы SAVE и LOAD могут использоваться также при работе с дисковым накопителем. При этом после них указываются: номер дисковода (≠ 1), номер диска (0 или 1 при двухдисковом дисковом), номер дорожки диска и номер сектора.

§ 2.2. Модификации языка бейсик

Бейсик имеет десятки модификаций, рассмотреть которые в одной книге едва ли возможно. Это и нецелесообразно, поскольку

достаточно ознакомиться с несколькими характерными версиями бейсика, начиная с простых (§ 2.1) и кончая расширенными, описанными ниже.

Версия бейсика систем подготовки программ (СПП) на базе микроЭВМ Электроника-ДЗ-28 содержит ряд дополнительных операторов, приведенных в табл. 2.6. Прежде всего они расширяют набор используемых математических функций.

Оператор PRINT TAB α обеспечивает перемещение места печати (или индикации) в строке на позицию с номером, задаваемым целой частью арифметического выражения, числа или переменной α. Этот оператор используется для табуляции табличных данных и печати простых графиков (см. § 2.5).

Оператор CALL (вызов) служит для вызова внешней подпрограммы с номером n. Внешняя подпрограмма должна завершаться оператором RTS (код 0511). Внешние программы загружаются в начале диалога и могут писаться в кодах, что существенно ускоряет вычисления.

Оператор ON (по) обеспечивает безусловный переход (ветвление) по адресу, заданному целой частью арифметического выражения, числа или переменной α (косвенная адресация).

Версия бейсика микроЭВМ FX-702P класса Pocket Computers содержит практически все операторы описанных выше версий. Дополнительные операторы этой версии приведены в табл. 2.7. Они наглядно иллюстрируют большие функциональные возможности, которыми могут обладать даже сверхминиатюрные ЭВМ класса Pocket Computers.

В состав этой версии входит ряд дополнительных операторов вычисления математических функций: ! (факториала), FRAC (выделения дробной части чисел), DEG и DMS (преобразования углов), RPC и PRC

Таблица 2.6

Дополнительные операторы расширенной версии языка бейсик для системы подготовки программ на базе микроЭВМ Электроника-ДЗ-28

Оператор	Функция	Пояснение
DEG(X)	$X \leftarrow 180X/\pi$	Перевод угла из радиан в градусы
RAD(X)	$X \leftarrow \pi X/180$	Перевод угла из градусов в радианы
TAN(X)	$\operatorname{tg} x$	Вычисление тангенса
ASN(X)	$\arcsin x$	Вычисление арксинуса
ACS(X)	$\arccos x$	Вычисление арккосинуса
HSN(X)	$\operatorname{sh} x$	Вычисление гиперболического синуса
HCS(X)	$\operatorname{ch} x$	Вычисление гиперболического косинуса
HTN(X)	$\operatorname{th} x$	Вычисление гиперболического тангенса
AHS(X)	$\operatorname{arsh} x$	Вычисление обратного гиперболического синуса
AHC(X)	$\operatorname{arch} x$	Вычисление обратного гиперболического косинуса
ANT(X)	$\operatorname{arth} x$	Вычисление обратного гиперболического тангенса
LGT(X)	$\log x$	Вычисление десятичного логарифма
EXT(X)	10^x	Вычисление функции 10^x
PRINT TAB α	—	Перемещение места печати (или индикации) в строке на позицию с номером, заданным целой частью арифметического выражения, числа или переменной α (от 0 до 101)
CALL n	—	Вызов внешней подпрограммы с номером n (от 1 до 7999)
ON α	—	Безусловный переход по адресу, заданному целой частью арифметического выражения, числа или переменной α

Дополнительные операторы расширенной версии языка бейсик ПЭВМ
FX-702P класса Pocket Computers

Оператор	Пояснение
X !	Вычисление факториала $x!$
FRAC X	Выделение дробной части числа x
DEG(X, Y, Z)	Преобразование угла, выраженного в градусах X, минутах Y и секундах Z, в угол, выраженный в десятичных долях градуса
DMS X	Преобразование угла, выраженного в градусах и их десятичных долях, в угол, выраженный в градусах, минутах и секундах с десятичными долями
RPC X, Y	Преобразование прямоугольных координат в полярные
PRC X, Y	Преобразование полярных координат в прямоугольные
RND(α , Y)	Округление численного значения арифметического выражения или числа α с погрешностью до 10^Y
CNT	Подсчет числа введенных чисел при статистических вычислениях
SDX	Стандартное отклонение x_i (смещенное)
SDY	Стандартное отклонение y_i (смещенное)
SDXN	Стандартное отклонение x_i
SDYN	Стандартное отклонение y_i
MX	Вычисление среднего значения x_i
MY	Вычисление среднего значения y_i
SX	Вычисление суммы x_i
SY	Вычисление суммы y_i
SX2	Вычисление суммы x_i^2
SY2	Вычисление суммы y_i^2
SXY	Вычисление суммы произведения $x_i y_i$
1 RA	Вычисление параметра a линейной регрессии $y(x) = a + bx$
LRB	Вычисление коэффициента b линейной регрессии $y(x) = a + bx$
COR	Вычисление коэффициента парной корреляции r
EOX y	Вычисление x по заданному y при линейной регрессии
EOY x	Вычисление y по заданному x при линейной регрессии
INP <V> AB	Присвоение символьной переменной AB значения символа <V> по оператору INP
\$ = "<V>"	Присвоение символьной переменной \$ значения символа <V>, содержащего до 30 знаков
B\$ = "<V>"	Присвоение символьной переменной B\$ значения символа <V>, содержащего до 7 знаков
C\$ = A\$ + B\$	Суммирование символов символьных переменных A\$ и B\$
LEN(C\$)	Вычисление числа символов в символьной переменной C\$
MID(m , n)	Выделение n символов символьной переменной \$, начиная с символа с номера m
A\$ = KEY	Присвоение клавише A свойств клавиши пользователя
VAC	Стирание всех ячеек памяти (обнуление всех переменных)
SAC	Стирание ячеек памяти статистических расчетов
CLR ALL	Стирание всех программ (от P0 до P9)
PASS"K"	Кодирование (засекречивание) программы кодом K (до 8 знаков)
GSB # n	Обращение к подпрограмме, помещенной в программную область P n
GOTO # n	Безусловный переход к программе, помещенной в программной области P n
GSB α	Обращение к подпрограмме с косвенной адресацией (номер строки задается арифметическим выражением α)
GSB # α	Безусловный переход с косвенной адресацией (адрес задается значением арифметического выражения α)
GOTO α	Безусловный переход к программе P n , номер которой n задается значением арифметического выражения α
GOTO # α	Безусловный переход к программе P n , номер которой n задается значением арифметического выражения α
IF N\$ = "<V>"	Условный переход по адресу n или к программе P n , если символьная константа N\$ равна символу <V>
THEN n , # n	
DEFM n	Высвобождение памяти под $10(n+1)$ переменных массивов, где $n = 0 \div 19$
CSR n	Перемещение начала индикации или печати на позицию n ($n = 0 \div 19$), оператор используется вместе с оператором PRT
PRT #.#.#.#	Задание формата чисел в форме с фиксированной запятой (знак # соответствует одной цифре)
PRT #.#.#.#↑	Задание формата чисел в экспоненциальной форме с плавающей запятой (знак # соответствует одной цифре)
SET N	Задание стандартной формы чисел с 10-ю знаками мантиссы

Оператор	Пояснение
SET E n	Задание экспоненциальной формы чисел с n знаками мантиссы ($n \leq 10$)
SET F n	Задание формы чисел с фиксированной запятой и n знаками после десятичной точки
WAIT n	Задание индикации с перерывом счета на время, равное $n \cdot 0,05$ с ($n \neq 0 \div 1000$)
STAT α, β	Проведение статистических расчетов с парой чисел X и Y, задаваемых значениями переменных X, Y, констант или арифметических выражений α и β
MODE 4	Задание углов в градусах
MODE 5	Задание углов в радианах
MODE 6	Задание углов в градах
SAVE #n <f>	Запись программы Pn с наименованием файла <f> на кассетный магнитофон
LOAD #n <f>	Ввод программы с наименованием файла <f> в область Pn с кассетного магнитофона
SAVE ALL "<f>"	Запись всех программ и данных с названием файла <f> на кассетный магнитофон
LOAD ALL "<f>"	Ввод всех программ и данных с названием файла <f> в микроЭВМ с кассетного магнитофона
LIST #n	Распечатка листинга программы Pn принтером
LIST #n "K"	Распечатка листинга засекреченной кодом "K" программы Pn принтером
LIST ALL	Распечатка всех программ (от P0 до P9) принтером
LIST V	Распечатка принтером значений переменной V
PUT "<f>" A, B	Запись файла данных (переменных A, B) с именем <f> на кассетный магнитофон
GET "<f>" \$, C	Вызов значений переменных \$, C файла с именем <f> с кассетного магнитофона
VER "<f>"	Контроль правильности считывания файла с именем <f> кассетным магнитофоном

(преобразования координат), округления, чисел с погрешностью до 10^6 , вычисления 17 статистических характеристик двумерного массива чисел (см. § 5.3).

В эту версию входит ряд операций с символьными переменными.

Символьной переменной называется переменная (обозначаемая любой латинской буквой со знаком \$ после нее), которая может принимать значение символа <V>. Например, оператор $B\$="ABC"$ присваивает символьной переменной B\$ символ ABC, который должен помещаться в кавычки. При исполнении оператора PRINT B\$ на индикацию или печать будет выведен символ — в нашем случае ABC.

Суммирование символьных переменных с помощью оператора вида $C\$=A\$+B\$$ означает простое суммирование символов: если $A\$="ABC"$, $B\$="DEF"$, то получим $C\$="ABCDEF"$.

Оператор LEN (C\$) обеспечивает подсчет числа знаков в символьной переменной. В нашем примере после исполнения этого оператора получим число 6.

Оператор MID (m, [, n]) *) выделяет n знаков символьной переменной \$, начиная с символа с номером m. Допустим, символьная переменная \$ = "STARTFINISHEND". Для выделения слова, выделенного полужирным шрифтом, имеющего $n=6$ букв и начинающегося

с символа F с номером $m=6$, надо дать команду MID (6,6). Так, если $A\$=MID(6,6)$, то символьной переменной A\$ будет присвоен символ FINISH.

Оператор $A\$=KEY$ обеспечивает присвоение клавише символьной переменной функции клавиши пользователя (key — клавиша). Тогда, используя операторы вида $IF A\$="<V>" THEN n$, можно начать выполнение программ со строки с номером n, присвоив переменной A\$ значение символа <V> (например, с помощью оператора INP (сокращение от INPUT), см. § 2.5).

Операторы стирания VAC, SAC и CLR ALL служат для обнуления всех переменных (VAC), стирания ячеек памяти при статистических расчетах (SAC) и общего стирания всех программ (CLR ALL). Операторы VAC и SAC могут вводиться в программу, оператор CLR ALL — служебный (выполняется в режиме записи программы MODE 1).

Оператор кодирования программ PASS (код) служит для присвоения программам специального кода (до 8 знаков) в режиме ввода программ — MODE 1. Не зная кода, невозможно вывести листинг программы или стереть программу оператором CLR (стирание возможно командой CLR ALL). Для вывода листинга код надо указать в кавычках после оператора LIST: n (n — номер строки) или LIST.

Таким образом, оператор обеспечивает защиту программы пользователя.

*) В квадратных скобках ставится необязательный элемент выражения.

Операторы GSB $\#n$ и GOTO $\#n$ служат для обращения к программной области Pn (напомним, что эти области 10 с n от 0 до 9). Оператор GSB $\#n$ применяется, если программная область Pn выполняет функции подпрограммы. При этом она должна завершаться оператором RET (сокращение от RETURN). Оператор GOTO $\#n$ обеспечивает вход в программную область Pn без автоматического возврата из нее.

Операторы косвенной адресации. Операторы GSB α , GSB $\# \alpha$, GOTO α и GOTO $\# \alpha$ обеспечивают обращение к подпрограмме или безусловные переходы к косвенной адресацией — адрес указывается целой частью арифметического выражения или значения переменной α .

Оператор DEFM n служит для задания массива из 10 ($n+1$) ячеек памяти. Число n задается от 0 до 19. При этом число шагов программы сокращается с 1680 до 80, т. е. увеличение n на 1 сокращает программу на 80 шагов.

Оператор CSR в составе оператора PRT (сокращение от PRINT) служит для смещения начала индикации на позицию n ($n = 0 \div 19$). Он аналогичен оператору TAB в других версиях бейсика.

Задание формата печати (индикации) обеспечивают операторы вида PRT $\# \# . \# \#$ или PRT $\# \# . \# \# \uparrow$. Каждый символ $\#$ до и после десятичной точки задает соответственно печать или индикацию одной цифры результата. Если последний не уместится в формат, то на печать или индикацию выводятся знаки $\#$. Символ \uparrow задает переход к индикации чисел в экспоненциальной форме (см. примеры ниже):

К перспективным версиям языка бейсик принадлежит и бейсик-80 — алгоритмический диалоговый язык высокого уровня для работы с новыми разработками отечественных персональных микроЭВМ и микропроцессорных систем [26]. В программы на языке бейсик-80 могут включаться подпрограммы, записанные на других языках: фортране, ПЛ/М-80, ассемблере. Бейсик-80 обеспечивает операции с целыми числами и числами с плавающей запятой с обычной и двойной точностью. Возможно представление чисел и в шестнадцатеричной форме.

Бейсик-80 имеет мощные средства для эффективного редактирования программ и работы с внешним периферийным оборудованием. Так, при работе с дисковыми накопителями бейсик-80 позволяет не только записывать и считывать информацию, но и читать и стирать оглавления файлов, создавать, переименовывать и изменять атрибуты без обращения к операционной системе. Бейсик-80 обеспечивает и работу с символьными (строчными) переменными.

Алфавит бейсик-80 соответствует общепринятому (см. § 2.1). В языке бейсик-80 различают директивы, операторы и функции.

Директивы (табл. 2.8) — команды, выполняемые непосредственно после их ввода с пульта. Они управляют программами или файлами. Большинство директив может использоваться в операторах программ. Однако следует помнить, что многие из них останавливают выполнение программы, обнуляют значения переменных или делают их неопределенными.

Операторы (табл. 2.9) выполняются по мере того, как они встречаются в программе.

Оператор	Индикация X=0,123456789	Индикация Y=12,3456789
PRT $\#$: X; Y	0	#
PRT $\# . \# \#$; X; Y	0.12	#.# #
PRT $\# \# . \# \# \#$; X; Y	0.123	12.345
PRT $\# . \# \# \# \uparrow$; X; Y	1.235E-01	1.125E01

Общее число знаков $\#$ после оператора PRT может быть до 10 включительно.

Операторы SET N, SET Eп и SET Fn задают формы представления чисел: SET N — стандартную форму с 10-ю знаками мантиссы, SET Eп — экспоненциальную форму с n знаками мантиссы. Оператором SET Fn задается форма чисел с фиксированной запятой и n знаками после нее. Оператор SET 5 обеспечивает округление последнего знака результата, а оператор SET 0 — сброс округления. В последнем случае происходит усечение результата.

Оператор задания паузы WAIT n обеспечивает по оператору PRINT паузу в вычислениях с выдачей результата на время $n \cdot 0,05$ с, причем n может задаваться от 0 до 1 000.

Смысл остальных операторов расширенной версии бейсика микроЭВМ FX-702P очевиден из пояснений, приведенных в табл. 2.7.

Они составляют большинство рабочих предписаний (команд) программ.

Функции (табл. 2.10) служат для образования арифметических и логических выражений совместно с директивами и операторами. Функции подразделяются на арифметические и строчные (для работы с символьными переменными).

В бейсике-80 21 директива, 39 операторов и 48 функций. Все они образуют инструкции данного языка.

Представление данных в бейсике-80. Бейсик-80 оперирует с арифметическими данными в виде констант и переменных. Константы с обычной точностью определяются суффиксом !, который ставится после арифметического значения, или с помощью буквы E, обозначающей представление константы в экспоненциальной форме с плавающей точкой (запятой). В последнем случае числа лежат в диапазоне от $\pm 1.2E-38$ до $\pm 3.4E+38$.

Директивы языка бейсик-80

Директива	Описание	Примеры
ATTRIB AUTO	Изменить атрибуты файла Задать автоматическую нумерацию строк с заданным приращением и номером начальной строки	ATTRIB:"F1:STAT,W" AUTO 25,500
CLEAR	Установить дополнительную память для строк (в байтах)	CLEAR 2000
CONT DELETE	Продолжить выполнение программы Уничтожить строку (или строки) в программе	CONT DELETE 700--875
DIR EDIT EXIT	Прочитать оглавление файлов на диске Изменить указанный оператор программы Вернуть управление операционной системе	DIR 1 EDIT 170 EXIT
KILL LIST LOAD MERGE	Уничтожить файл на диске Вывести строку (или строки) программы Загрузить указанный файл с диска Соединить указанный программный файл с текущей программой	KILL ":F1:STAT" LIST 300--400 LOAD ":F1:DATE S" MERGE ":F1:DATES"
NEW	Уничтожить текущую программу и очистить переменные	NEW
PRUN	Выполнить программу, хранящуюся в ПЗУ	PRUN 4 E00H
RENAME RENUM RUN SAVE TRON TROF WIDTH	Изменить имя файла Изменить номера строк программы Начать выполнение программы Записать программу или файл на диск Включить трассировку Выключить трассировку Изменить ширину строки	RENAME ":F1:SOUP" TO ":F1:NUTS" RENUM 'RUN SAVE ":F1:INVEN" TRON TROF WIDTH 80

Таблица 2.9

Операторы языка бейсик-80

Оператор	Описание	Примеры
CLOSE DATA	Закрывать один или более файлов Задать значения, которые могут быть считаны оператором READ	CLOSE 3 DATA 9, 0, ON
DEF	Определить написанную пользователем функцию	DEF FNRT (R1, R2) = R1*R2/(R1+R2)
DEFDBL	Определить переменные, имена которых начинаются с указанной литеры как переменные с плавающей точкой с двойной точностью	DEFDBL R — Z
DEFINT	Определить переменные, имена которых начинаются с указанной литеры как целочисленные переменные	DEFINT I — N
DEFSNG	Определить переменные, имена которых начинаются с указанной литеры как переменные с плавающей точкой с обычной точностью	DEFSNG B — H, X
DEFSTR	Определить переменные, имена которых начинаются с указанной литеры как строчные переменные	DEFSTR K — O
DEFUSR DIM	Ввести подпрограммы на другом языке Зарезервировать память для переменных массива (массивов)	DEFUSR — 4E00H DIM A (25,10,30)
END ERROR	Закончить программу Имитировать ошибки с данным номером	END ERROR 12

Оператор	Описание	Примеры
FIELD	Зарезервировать память для буфера файла прямого допуска	FIELD # 3,20, ASA □
FOR...TO...NEXT...STEP	Создать цикл ...	FOR I = 1 TO 5 STEP. 5 NEXT I
GET	Прочитать данные на диске файла	GET #2,4
GOSUB	Выполнить подпрограмму	GOSUB 500
GOTO	Выполнить безусловный переход по указанному номеру строки	GOTO 1000
IF...THEN...ELSE	Выполнить оператор, если указанное выражение истинно; если ложно — выполнить второй оператор	IF A > B THEN 250 ELSE PRINT "END"
INPUT	1. Ввести данные с терминала 2. Прочитать данные из последовательного файла	INPUT A, B, C INPUT #1, A □, B □
LET	Присвоить значение переменной	LET X = 100
LINE INPUT	Ввести целую строку из последовательного файла	LINE INPUT A □
LSET	Выводить указанные строки по левому краю	LSET A □ = B □
ON ERROR	Перейти к строке с заданным номером, если появилась ошибка	ON ERROR GOTO 900
ON GOTO	Перейти к строке с номером, заданным значением выражения X	ON X GOTO 10, 50, 80
ON GOSUB	Перейти к подпрограмме, номер первой строки которой задается значением выражения X	ON X GOSUB 100, 200, 500
OPEN	Создать файлы последовательного или прямого доступа	OPEN "R", #3, "F1:PERSON"
OPTION BASE	Присвоить начальные значения элементам всех массивов (0 или 1)	OPTION BASE 0
OUT	Записать указанное целое число в устройство ввода-вывода	OUT 00F0, 12
POKE	Записать байт в указанное место памяти	POKE 0A077, 12
PRINT	1. Вывести значения переменных на терминал 2. Завести данные в последовательный файл на диске	PRINT A, B, C PRINT #4, A □, B □
PRINT USING	Представить текст в соответствии с данным форматом	PRINT USING "□ □ # #, # #; 17.45; 24.67"
PUT	Занести данные в файл прямого доступа на диске	PUT #3; A □, B □, C □
RANDOMIZE	Инициализировать генератор случайных чисел	RANDOMIZE
READ	Присвоить значения переменным программы из списка оператора DATA	READ A, K1, L, Z
REM	Напечатать комментарий в тексте программы	10 REM START
RESTORE	Установить заново указатель для чтения данных из оператора DATA	RESTORE
RESUME	Возобновить выполнение программы после обнаружения ошибки	RESUME
RETURN	Возвратить управление оператору, следующему за последним GOSUB	RETURN
RSET	Выводить указанные строки по правому краю	RSET L □ = MK □
STOP	Остановить выполнение программы	STOP
SWAP	Провести обмен значениями для двух переменных одинакового типа	SWAP A1 □, B2 □
WAIT	Остановить выполнение программы до тех пор, пока при сравнении не получится значение «Истинно»	WAIT 1, 04H, 0AH
LEN	Определить число символов в указанной строке	LEN (X □)

Оператор	Описание	Примеры
LOC	Определить текущий номер записи в файле прямого доступа X или количество прочитанных или записанных секторов в последовательном файле X	LOC (#X)
LOF	Вычислить число записей в файле прямого доступа или число секторов с данными в последовательном файле X	LOF(X)
LOG	Вычислить натуральный логарифм от X	LOG(X)
MID□	Указать J литер, начиная с I-й, в строке A□	MID□ (A□, I, J)
MKI□	Преобразовать 2-байтную строку в целое значение	MKI□ (A□)
MKS□	Преобразовать 4-байтную строку в значение с плавающей точкой с обычной точностью	MKS□ (A□)
MKD□	Преобразовать 8-байтную строку A□ в значение с плавающей точкой с двойной точностью	MKD□ (A□)
OCT□	Преобразовать систему восьмеричных цифр в эквивалент десятичного аргумента	OCT□ (X)
PEEK	Указать отдельный байт памяти по адресу X	PEEK(X)
POS	Указать позицию курсора после последнего PRINT	POS(DUMMY ARGUMENT)
RIGHT	Вывести I% правых литер в строке	RIGHT (X□, I%)
RND	Выдать случайное число обычной точности со значением между 0 и 1	RND
SGN	Указать знак X	SGN(X)
SIN	Вычислить синус	SIN(X)
SPACE□	Вывести I% пробелов	SPACE□ (X)
SPC	Указать строку X% пробелов	SPC(X%)
SQR	Вычислить корень квадратный из X	SQR(X)
STR□	Преобразовать строку десятичных цифр в десятичный аргумент X	STR□ (X)
STRING□	Вывести первую литеру строки A□, повторенную Y% раз	STRING□ (Y%, A□)
TAB	Начать вывод с позиции X дисплея или принтера	TAB(X)
TAN	Вычислить тангенс X	TAN(X)
USR	Обратиться к подпрограмме пользователя от 0 до 24	AX = USR I2(A1, A2)
VAL	Преобразовать числовое значение X□ аргумента	VAL (X□)
VARPTR	Указать адрес-памяти X	VARPTR(X)

Таблица 2.10

Функции языка бейсик-80

Функция	Описание	Примеры
ABS	Выдать абсолютное значение X	ABS(X)
ASC	Выдать код КОИВ для первой литеры указанной строки	ASC (A□)
ATN	Вычислить арктангенс (в радианах) X	ATN(X)
CDBL	Вывести значение X с удвоенной точностью с плавающей запятой	CDBL(X)
CHR□	Вывести литеру, соответствующую коду КОИВ	CHR□ (X)
CINT	Округлить X до целого числа	CINT(X)
COS	Вычислить косинус (в радианах) X	COS(X)

Функция	Описание	Примеры
CSNG	Вывести значение X с обычной точностью с плавающей точкой	CSNG(X)
CVD	Преобразовать 8-символьную строку в число с двойной точностью	CVD(X#)
CVI	Преобразовать 2-символьную строку в целое число	CVI(X%)
CVS	Преобразовать 4-символьную строку в число с обычной точностью	CVS(X!)
DSKF	Указать число свободных 128-байтных блоков на указанном диске	DSKF(I)
EOF	Выдать указание (да — 1, нет — 0) о том, достигнут ли конец файла при чтении последовательного файла данных	A=EOF (#X)
ERL	Указать номер строки, в которой произошла последняя ошибка	ERL
ERR	Указать код последней ошибки	ERR
EXP	Вычислить значение e^x	EXP(X)
FIX	Вычислить усеченное целое значение X	FIX(X)
FRE	Определить число байт, оставшихся в памяти, при задании фиктивного аргумента X или X□	FRE(X)
HEX□	Представить целочисленный аргумент X% (от —32768 до 32767) в 16-ричном виде	HEX□ (X%)
INP	Вывести значение байта из устройства ввода-вывода, указанного выражением в скобках	INP(X)
INSTR□	Указать позицию, заданную X□, в строке Y□	INSTR□ (X□, Y□)
INT	Вычислить наибольшее целое значение, меньшее или равное значению (X)	INT(X)
LEFT□	Вывести X литер строки A□, начиная с ее левого конца	LEFT (A□, X)

Примеры ввода констант: —1.414! (число —1,414); 5.26E—12 (число $5,26 \cdot 10^{-12}$).

Константы с плавающей точкой с двойной точностью определяются суффиксом # или буквой D.

Диапазон значений чисел при этом расширяется от $\pm 2.2E-308$ до $\pm 1.7E+308$.

Примеры ввода констант с двойной точностью: —1.2345678912D15 (число $-1,2345678912 \cdot 10^{15}$); —12,3# (число —12,3).

Арифметические переменные бывают трех типов: целые; с плавающей точкой с обычной точностью и с двойной точностью. Они представляют значения чисел, которые могут изменяться в процессе выполнения программы. Арифметические переменные языка бейсик-80 могут изображаться одной или двумя буквами, за которыми может следовать определяющий их тип суффикс. Первый элемент обозначения переменных обязательно должен быть буквой, второй — любым алфавитно-цифровым знаком. Если имя переменной содержит более чем две литеры (кроме суффикса, определяющего тип), остальные литеры имени переменной игнорируются.

До присвоения переменным численного значения оно задается равным нулю. Директивы NEW, RUN, CLEAR, LOAD и MERGE обнуляют все переменные.

С помощью операторов DEFINT, DEFSNG или DEFDBL можно определить ряд имен для обозначения арифметических переменных указанных типов. Группы переменных, которые начинаются с определенных букв, могут быть заданы с помощью этих операторов в

следующем виде:

DEFXXX M [N],

где N — любая буква от A до Z, M — любая буква от N до Z. При этом все они задаются как переменные одного типа.

По умолчанию все переменные имеют представление с обычной точностью. Если какие-то переменные должны быть иного типа, их следует определить в начале программы. Определители имеют вид знаков: % для целочисленных переменных, ! для переменных с обычной точностью и # для переменных с двойной точностью.

Строчные или символьные константы представляют собой группу знаков, заключенных в кавычки, например:

"NEWTON METHOD" или "PI=3.14159".

Внутри этих переменных нельзя использовать кавычки. Число знаков (длина строки) может доходить до 255.

Строчные переменные — переменные, принимающие по ходу выполнения программы символичные значения. Имя строчной переменной задается одним или двумя знаками: первый обязательно буква, второй — любой. После них следует символ□, определяющий символьную переменную.

Например, переменная A1□NEWTON METHOD имеет имя A1□ и значение символа NEWTON METHOD.

Преобразование данных из одного типа в другой в бейсике-80 обеспечивается с по-

мощью ряда специальных функций (см. табл. 2.10).

Массивы (одномерные и двумерные) в бейсике-80 задаются, как и в других версиях языка бейсик, именем переменной и номерами строк и столбцов в скобках: например, A2(3,4) задает переменную массива A2 с индексами 3 и 4. Задание памяти под массив обеспечивается оператором DIM. В строчных переменных массивов необходимо резервировать и число знаков в строке. Например, оператор DIM A X(10,20,25) резервирует память под строчную переменную из массива AX.

Приоритет выполнения операций в бейсике-80.

Операция	Пример
1. Выражение в скобках	(A + B)
2. Возведение в степень	A ^ B
3. Отрицание (обозначается знаком —)	-A
4. Умножение (*) и деление (/)	A*B или A/B
5. Целочисленное деление (знак \) с преобразованием делимого, делителя и результата в целые числа	A \ B
6. Модуль числа	AMOD
7. Сложение и вычитание	A + B или A - B
8. Операции отношения	A > B и др.

Самый низкий приоритет у логических операций. Они выполняются в порядке их наименования: НЕ, И, ИЛИ, исключаящее ИЛИ, логическая импликация и логическая эквивалентность.

Арифметические и логические выражения в бейсике-80 строятся по правилам, типичным для других версий (они описаны выше).

Специальные функции языка бейсик-80 — такие функции, которые характерны для данной версии бейсика и имеют те или иные отличия от функций бейсика, описанных ранее.

Арифметическая функция CDBL(X) изменяет тип выражения X на представление с двойной точностью с плавающей точкой. В этом формате результат выдается с точностью до 16 десятичных цифр.

Пример.

```
10 A# = 1/3
20 B# = CDBL(1/3)
30 PRINT A#, B#
40 END
.333333 .3333333333333333
```

Строчная функция CERQ (Выражение) приводит выражение к целому числу в интервале от 0 до 255. Арифметическая функция CINT(X) округляет выражение (X) до целого значения.

Пример.

```
10 INPUT A, B, C
20 PRINT (CINT(A + B + C) / 3)
30 END
RUN
? 45, 24, 77
49
```

Функция CSNG(X) вычисляет выражение X с обычной точностью (до 7 десятичных позиций).

Пример.

```
10 A# = 1.DO/3.DO
20 PRINT CSNG(A#) / 2, A# / 2
RUN
.1666667 .1666666666666667
```

Функции CVI, CVS, CVD преобразуют строки в числа: CVI — 2-символьную строку в целое число, CVS — 4-символьную строку в число с обычной точностью и CVD — 8-символьную строку в число с двойной точностью. Они используются для восстановления числовых значений из поля буфера

файла прямого доступа и являются обратными функциями MKIQ, MKSQ и MKDQ.

Функция INP (Выражение) вводит значение байта со входного порта, номер которого указан целочисленным выражением (от 0 до 255).

Функция INPUTQ (Выражение; Номер файла) читает заданное количество символов из указанного номером файла.

Пример. В программе

```
10 OPEN "1" #2, ":C1"
20 PRINT INPUTQ(10,2)
30 CLOSE #2
40 END
RUN
123456789 (BK)
123456789
```

строка 20 печатает первые 10 символов файла #2, а оператор CLOSE закрывает файл.

Функция INSTR ищет первое вложение второй строки BQ в первой строке AQ и определяет порядковый номер второй строки. Она формирует 0 при трех условиях: если I больше длины AQ, если BQ не может быть найдена и если AQ не содержит символа. Вид записи функции:

INSTR ([I,] Строчное выражение AQ, Строчное выражение BQ).

Пример.

```
10 AQ = "RANDOM NUMBER
SUBROUTINE"
20 BQ = "R"
30 PRINT INSTR(AQ, BQ)
40 PRINT INSTR(2, AQ, BQ)
50 END
RUN
```

1
13

Функция INT(X) отбрасывает наибольшее целое значение, меньшее или равное указанному выражению или числу X. Например, если $B = \text{INT}(15,021)$, то распечатка B дает значение 15.

Строчная функция LEFT \square (Строчная переменная. Выражение) находит значение строки X \square и определяет l левых символов.

Пример.

```
10 X $\square$  = "WHITE, SMITH, JONES,  
BLACK, GREEN"  
20 Y $\square$  = LEFT $\square$  (X $\square$ , 11)  
30 PRINT Y $\square$   
RUN  
WHITE, SMITH
```

Строчная функция LEN(X \square) подсчитывает количество символов строки X \square , включая непечатаемые символы и пробелы.

Функция LOC в файле прямого доступа определяет текущий номер последней записанной или прочитанной записи. В последовательном файле функция LOC определяет число секторов, прочитанных или записанных со времени выполнения последнего оператора OPEN. Функция записывается в виде

LOC (Номер файла)

Пример.

```
10 OPEN "R", #3,:F1:RANDOM  
20 GET #3, 44  
30 PRINT LOC(3)  
40 END  
RUN  
44
```

Функция LOF (Номер файла) вычисляет число записей в файле прямого доступа, а в файле последовательного доступа вычисляет число секторов с данными (128 байт на сектор) в файле.

Функция MID \square (X \square , I [, J]) проверяет строку X \square и выделяет ее правые символы, начиная с позиции, заданной значением I. Здесь I и J — целые числа от 0 до 255. Если указан аргумент J, определяется J символов, начиная с позиции I. Если I больше, чем LEN(X \square), то MID \square формирует пустую строку. Если J больше числа символов в X \square справа от I-й позиции или J не указано, функция MID \square формирует остаток строки.

Пример.

```
10 X $\square$  = "JOHN J. JONES"  
20 PRINT MID $\square$  (X $\square$ , 10, 3)  
30 END  
RUN  
ONE
```

Функция MKI \square , MKS \square и MKD \square преобразовывают данные как числовые значения в 2-, 4- или 8-байтные строки соответственно. MKI \square используется для преобразования целого значения, MKS \square — для преобразования значения с плавающей запятой с обычной точностью и MKD \square — для преобразования значения с плавающей запятой с двойной точностью.

Функция OCT \square (Выражение) определяет строку восьмеричных цифр, представляющую значение целого аргумента (выражения).

Функция PEEK (Выражение) читает один байт памяти в указанном месте, а функция POKE (см. далее) записывает один байт в указанное место памяти. Например:

```
PRINT PEEK (OFABH)  
200
```

где 200 — прочитанный байт из места OFABH памяти.

Функция POS (Целое число или Выражение) указывает позицию курсора после последнего оператора PRINT. В записи POS(I) аргумент I является фиктивным, самая левая позиция курсора соответствует I=1.

Функция RANDOMIZE запрашивает ввод нового случайного числа, которое является начальным для функции генерации случайных чисел с равномерным распределением RDN. Если функция RANDOMIZE не используется, то одна и та же последовательность случайных чисел будет повторяться каждый раз. Функция RANDOMIZE записывается в виде

RANDOMIZE (Выражение)

причем начальное случайное число задается выражением (либо константой или переменной).

Функция RIGHT \square (Строка, Целое I) определяет I правых символов строки X \square . Если I=0, выдается пустая строка, а если I превышает длину строки X \square , то выдается вся строка.

Пример.

```
10 A $\square$  = "JOHN J. JONES"  
20 X $\square$  = RIGHT $\square$  (A $\square$ , 8)  
30 PRINT X $\square$   
40 RUN  
J. JONES
```

Функция SPACE \square (Целочисленное выражение) формирует в строке пустой интервал, состоящий из пробелов, число которых задается значением целочисленного выражения. Функция SPS (Целое число) совместно с оператором PRINT также формирует пустой интервал из N пробелов (N — целое число).

Функция STRING \square (Целое, Целое) определяет строку, в которой заданный символ повторяется указанное число раз. В записи STRING \square (Целое, Строчное выражение) заданное число раз повторяется первый символ строчного выражения. Функция STR \square (Выражение) формирует строку десятичных знаков, которая представляет значение целочисленного выражения.

Пример.

```
10 FOR I=0 TO 9  
20 A $\square$  = A $\square$  + MID $\square$  (STR $\square$  (I), 2)  
30 NEXT I  
40 PRINT A $\square$   
RUN  
0123456789
```

Функция USRN задает обращения к определенному пользователем подпрограммам, записанным на языке ассемблера, Фортран-80 или ПЛ/М-80. Оператор DEF USRN определяет начальный адрес соответствующей подпрограммы USRN. Аргумент N может быть любым целым от 0 до 24; если он не указан, полагается N=0. Символ типа обозначает тип результата. Если обозначение типа отсутствует, результат определяется как целое число в регистрах H и L микропроцессора.

Функция VAL (Строка) определяет численное значение строки.

Пример.

```
10 INPUT A$
20 A1=VAL (A$)*52
30 PRINT A1
   RUN
   ? 4 (т. е. A$=4)
208 (т. е. имеем 4*52)
```

Специальные директивы и операторы.

Под специальными директивами и операторами языка бейсик-80 подразумеваются специфические именно для него директивы и операторы. Ниже они даны в алфавитном порядке.

Директива ATTRIB изменяет атрибуты файлов на дисках или защищает любой файл от записи, стирания или изменения имени. Защита файла выполняется директивной в виде

ATTRIB:F Номер дисковогода, "Имя файла", "W1"

Запись, стирание и изменение ранее защищенного файла обеспечивается директивой

ATTRIB:F Номер дисковогода, "Имя файла", "W0"

Директива AUTO обеспечивает автоматическое присвоение номера каждой строке программы. Она вводится в виде

AUTO Приращение номера [, Номер первой строки]

Так, директива AUTO 50, 300 задает номера строк с приращением 50, начиная со строки 300, т. е. строки будут иметь номера 300, 350, 400, 450 и т. д. По умолчанию, т. е. при вводе только директивы AUTO, приращению и номеру первой строки присваиваются значения 10. Если задаваемый директивой AUTO номер строки имеется в программе, он отмечается знаком *.

Директива CLEAR обнуляет все переменные и массивы, определения функций и закрывает файлы. Символьные переменные при этом становятся пустыми строками. Эта директива может записываться в виде

CLEAR [Выражение [Адрес]]

Численное значение выражения указывает пространство для строк в байтах. Например, директива CLEAR 2000, 0D000H резервирует (высвобождает) 2000 байт для строк и не позволяет использовать адреса памяти выше 0D000H.

Директива CLOSE [Номер] закрывает файл с заданным номером. Если аргумент (номер) не указан, директива CLOSE за-

крывает все файлы. Все файлы закрывают также директивы END, NEW, MERGE, LOAD, RUN (без F).

Директива DEF FN(X) задает функцию пользователя. Здесь FN — имя функции, X — аргумент (функция может быть с любым числом аргументов).

Директива DELETE HC, где HC — номер строки, уничтожает одну строку с заданным номером, директива DELETE —HC уничтожает строку с номером HC и все предшествующие ей строки. Директива DELETE HC1, HC2 уничтожает строки с номерами от HC1 до HC2.

Директива DIR воспроизводит имена, число блоков и длины (в байтах) файлов, сохраняемых на указанном после нее диске.

Оператор END останавливает выполнение программы и закрывает все файлы. Переменные при этом сохраняют свои значения. Оператор STOP действует как END, но файлы не закрываются и при его исполнении является сообщение

BREAK IN HC

(т. е. останов в строке HC). Оператор END рекомендуется ставить в конце программы, хотя это не обязательно.

Оператор ERROR "Выражение" имитирует происхождение ошибки. Если ERROR выполняется с номером, который не соответствует сообщению об ошибке, будет напечатано сообщение

UNPRINTABLE ERROR

Оператор FIELD# «Номер файла» используется для назначения памяти в одном из шести 128-байтных буферов файлов прямого доступа.

Для чтения определенной записи из дискового файла в 128-байтный буфер служит оператор

GET[#] Номер файла [, Номер записи]

Оператор условных переходов в бейсике-80 может иметь вид

IF Выражение THEN Инструкция ELSE Инструкция

Например, оператор

50 IF X=0 THEN A=5 ELSE A=20

означает, что если X=0, то A=5, иначе A=20.

Оператор

KILL Строчное выражение

служит для исключения из дисковой памяти программы. Он также уничтожает заголовок программы в оглавлении.

Оператор INPUT помимо обычного применения (задания численных значений переменной или переменным списка) используется в составном операторе LINE INPUT. Последний читает целую строку данных и присваивает ее указанной строчной переменной в качестве значения. Данные могут быть прочитаны из файла на диске или введены

пульты. В последнем случае оператор LINE INPUT задается в форме

LINE INPUT [Строчное выражение;] Строчная переменная;

При этом оператор обеспечивает печать строчного выражения в качестве напоминания и ждет указания строчной переменной. При вводе данных с файла оператор записывается в виде

LINE INPUT # Номер файла, Строчная переменная

Директива MERGE читает программу с диска и соединяет ее с текущей программой без изменения строк каждой из программ. Если программы имеют строки с одинаковыми номерами, то введенные с диска строки замещают строки текущей программы. Директива MERGE стирает все переменные. Она записывается в виде

MERGE Строчное выражение

Директива NEW стирает всю текущую программу и обнуляет все переменные.

Оператор ON ERROR GOTO HC передает управление на строку с номером HC, если возникает ошибка. Оператор

ON Выражение GOTO HC1, HC2, ...

передает управление строкам HC1, HC2 и т. д. по результатам вычисления выражения (либо значения переменной). Оператор

ON Выражение GOSUB HC1, HC2, ...

обеспечивает обращение к подпрограммам, начинающимся со строк HC1, HC2 и т. д., по результатам вычисления выражения. Таким образом, операторы ON GOTO и ON GOSUB осуществляют косвенную адресацию.

Оператор

OPEN Тип, [#] Номер файла, Имя файла

делает файл операционной системы доступным для программы. Тип файла указывается знаками: R — для файла с прямым доступом, l — для последовательного вводного файла и 0 — для последовательного выводного файла. Номер файла — целое число от 0 до 6, перед которым ставится знак #. Одновременно можно открыть только 6 файлов.

Директива OPTION BASE N (где N=0 или 1) задает индексирование массивов с единицы (N=1) или с 0 (N=0).

Оператор

OUT Номер порта, Выражение

записывает значение выражения в виде целого числа от 0 до 255 в указанный номер порта.

Оператор

POKE Ячейка, Выражение

помещает значение выражения (округленное до целого числа) в указанную ячейку памяти. До использования этого оператора нужно использовать директиву CLEAR, чтобы перезагрузить свободную память.

Оператор PRINT выводит на печать значения выражений или текст. Литерные строки

PRINT USING [# Номер файла,] Строка формата; Выражение, ...

обеспечивает вывод при специальных форматах. Строка формата определяет пробелы и дополнительные символы (*□) при печатании числовых полей или определяет части данной строки при печатании строчных полей. Необязательный номер файла определяет последовательный дисковый файл.

Для оператора PRINT USING со строчными полями существуют два формата: «!>» определяет, что печатается первый символ каждой указанной строки; N пробелов определяют, что печатаются первые 2+N символов (если используется обратная косая черта без пробелов, то печатаются 2 символа). Если поле содержит больше символов, чем принимаемая подстрока, лишние символы отбрасываются; если символов меньше — вставляются пробелы, чтобы заполнить поле.

Пример.

```
10 X□ = «ONE»
20 Y□ = «TWO»
30 PRINT USING «!>»; X□; Y□
40 PRINT USING 5; X□; Y□
RUN
OT
ONE TWO
```

Число цифр, выводимых на печать оператором PRINT USING, указывается символами #, между которыми может стоять знак «.» или «.». Число знаков # до знака «.» или «.» задает вывод целой части, а после этих знаков — дробной части.

При необходимости числа округляются, например:

```
PRINT USING «# #, # #»; 41.287
41.29
PRINT USING «# #.# #»; 71/100
0.71
```

Если число, которое выводится на печать, мало, чтобы заполнить все указанные цифры слева от десятичной точки, вставляются пробелы, заполняющие все поле. До 5 цифр можно напечатать для целого числа, до 7 — для числа с плавающей точкой с обычной точностью и до 15 — для числа с плавающей точкой с двойной точностью.

Знак + может использоваться в начале или в конце поля символов формата. Он печатается в соответствующем месте. Знак — используется справа поля символов формата, чтобы вызвать печать закрывающего знака «минус», если число отрицательное (если число положительное, печатается пробел). Если ни +, ни — не включены в формат, отрицательное число печатается со знаком —.

Примеры.

```
PRINT USING «+ #.# #»
4.89; -2.6689
+4.89; -2.67
PRINT USING «# #.# #»; -5.8; 96.2;
-5.8
96.2
```

С помощью символов ** предшествующие пробелы заполняются звездочками:

```
PRINT USING «**# #.#»; 4.8; 243.3;  
***4.8 *243.3
```

Символы □ □ добавляют один знак □ слева от формируемого числа:

```
PRINT USING «□ □ # #.# #.#»;  
—48.28; 364.90;  
— □48.28□364.90
```

Знаки □ □ определяют пространство для двух дополнительных символом, но добавленный символ □ использует одну позицию.

Символы ↑↑↑↑ задают формат числа в экспоненциальной форме. Стрелки помещаются после цифровых символов формата. Символ % выдается перед числом, если оно не входит в формат:

```
PRINT USING «# #.# #.#»; 40.48;  
99.997;  
40.48 %99997
```

Директива
PRUN Адрес

обеспечивает выполнение программы, хранящейся в ППЗУ. Адрес программы — целое число.

Оператор

PUT Номер файла, [Номер записи]

передает данные из определенного файла в указанную запись диска.

Оператор REM используется для ввода в программу программных комментариев, которые выводятся на печать только при выводе листинга программы. Комментарии записываются непосредственно после оператора REM, например:

```
10 REM ЧИСЛЕННОЕ  
ИНТЕГРИРОВАНИЕ МЕТОДОМ  
СИМПСОНА
```

Директива

```
RENAME "Старое имя файла" TO  
"Новое имя файла"
```

изменяет имя указанного файла на новое с изменением ссылки в оглавлении.

Директива

```
RENUM [Новый номер] [, Старый номер]  
[, Приращение]
```

переупорядочивает номера строк в программе независимо от того, появляются ли они в начале строки или являются аргументом в операторе GOTO. В директиве можно указать три необязательных аргумента: новый номер первой перенумерованной строки, старый номер перенумерованной строки и приращение строк. Если аргументы не указаны, первой строке присваивается номер 10 и приращение принимает значение 10.

Пример.

```
20 INPUT A  
40 PRINT «NEW»; A
```

15 RANDOMIZE

5 PRINT CHR(12)

52 A1=A*RND

58 A2=INT(A)

RENUM

LIST:

10 PRINT CHR(12)

20 RANDOMIZE

30 INPUT A

40 PRINT «NEW»; A

50 A1=A*RND

60 A2=INT(A1)

Оператор RESUME возобновляет выполнение программы после управления ошибкой, которая обнаруживается. Выполнение программы начинается с указанной строки. Если номер строки не указан, выполнение программы происходит с оператора, в котором была обнаружена ошибка. Оператор

RESUME [Номер строки] NEXT

обеспечивает выполнение оператора, следующего за оператором с ошибкой.

Директива RUN обеспечивает выполнение программы, начиная с наименьшего номера строки. Если за оператором RUN следует строчная переменная, обеспечивается поиск файла с заданным ею именем, загрузка его в память и выполнение программы, начиная с наименьшего номера строки.

Ввод символа ↑C; в том числе вместо данных по оператору INPUT, останавливает выполнение программы и вызывает печать сообщения BREAK IN HC (где HC — номер строки, на которой прервано выполнение программы).

Директива

SAVE Строчное выражение [A]

заносит текущую программу на диск с указанным строчным выражением именем файла. Если файл с таким именем уже есть на диске, он будет заменен новым.

Оператор

SWAP Переменная 1, Переменная 2

обеспечивает обмен значениями двух переменных: переменной 1 и переменной 2.

Директивы TRON и TROF соответственно включают и выключают трассировку программы, т. е. обеспечивает пооператорное выполнение программы.

Директива

WAIT Номер порта, Байт маски, Байт сравнения

предписывает принять поступившие с указанного порта байты. Эти байты проверяются по байту маски, который является целым выражением (от 0 до 255). Если байт сравнения не указан, он задается равным 0. Директива WAIT выполняет операцию исключающее ИЛИ над состоянием порта и байтом сравнения, а результат и байт маски подтверждаются операцией И. Выполнение продолжается до тех пор, пока не будет получена ненулевая величина.

WIDTH Выражение

задает ширину печатаемых на консоли строк к заданной выражением величине (от 15 до 255). Если выражение не указано, ширина строк задается с 72 знаками.

Особенности ввода и редактирования программ на бейсике-80. Бейсик-80 имеет ряд дополнительных возможностей для ввода и редактирования программ. Так, используя директиву AUTO, можно автоматически нумеровать строки, начиная с заданного номера с заданным приращением. После ввода директивы AUTO бейсик-80 формирует номер строки и ждет ввода операторов. Если ввод их закончен символом возврата каретки BK, интерпретатор бейсик-80 формирует номер следующей строки и т. д. Ширина поля печати равна 72 символам, но может изменяться директивой WIDTH.

Бейсик-80 обеспечивает редактирование программ: стирание и замену текстов, отдельных строк и предложений, коррекцию символов и предложений, коррекцию символов в операторах, объединение строк и др. Он имеет также возможности для работы с дисковым накопителем. Бейсик-80 включает в себя два типа операций ввода-вывода файлов на диске: последовательный и с прямым доступом. Последовательный позволяет читать и писать файл от начала до конца; а с прямым доступом — указывать, где в файле выполняется чтение и запись. Операторы работы с файлами были описаны выше. Более подробные сведения о редактировании программ и работе с периферийным оборудованием можно получить из [26] и технической документации на вычислительные средства, использующие версию бейсик-80.

Данные о некоторых других версиях бейсика. В некоторых версиях бейсика, например, бейсик-плюс для серии отечественных мини-ЭВМ (включая ЭВМ СМ-4) и бейсик-3А-TDM для ЭВМ Электроника-ДЗ-28 имеется ряд дополнительных операторов. Прежде всего, к ним относятся операторы выполнения матричных операций.

Оператор чтения элементов матриц MAT READ A, B, C, ... обеспечивает чтение элементов матриц A, B, C, ... последовательно по строкам. Массивы элементов матриц предварительно задаются операторами DIM и DATA:

```
10 DIM A(3,3)
20 DATA 1,2,3,4,5,6,7,8,9
30 MAT READ A
```

Оператор ввода элемента матрицы с пульта MAT INPUT A обеспечивает ввод элементов матрицы A построчно с клавишного пульта.

Оператор вывода матрицы MAT PRINT A, B, C, ... обеспечивает вывод массива элементов матриц A, B, C, ... построчно и последовательно, т. е. вначале выводятся элементы матрицы A, затем B и т. д.

Оператор задания нулевой матрицы MAT A=ZER формирует матрицу A, у которой все элементы нулевые.

Элементы матрицы MAT A=CON создаются элементами MAT A=CON создает матрицу A, у которой все элементы имеют значения 1.

Оператор задания единичных элементов a_{kk} MAT A=IDM создает матрицу A, у которой элементы $a_{kk}=1$, а остальные 0. Если матрица A квадратная, то этот оператор создает единичную матрицу.

Оператор транспонирования MAT B=TRN(A) обеспечивает транспонирование матрицы A, т. е. получение матрицы B, у которой элементы $b_{ji}=a_{ij}$. Матрицы A и B не должны совпадать, а их размерности должны соответствовать возможности транспонирования. В противном случае ПЭВМ выдает указание об ошибке.

Оператор обращения матриц MAT C=INV(A, B) или MAT B=INV(A). Форма записи зависит от версии бейсика. В первом виде исходной является квадратная матрица C и должны быть зарезервированы два служебных массива A и B. После выполнения оператора MAT C=INV(A, B) обращенная матрица помещается на место исходной, а элементу A(0) присваивается значение определителя D исходной матрицы C. Во втором виде оператора исходной является матрица A и формируется матрица $B=A^{-1}$. Определитель при этом получается как значение специальной переменной DET, которое может присваиваться обычной переменной (например, D=DET).

Оператор умножения матриц MAT C=A*B формирует матрицу C, получаемую умножением матриц A и B. Матрицы A, B и C не должны совпадать. Особенности выбора размерности матриц при умножении описаны в § 3.4.

Операторы арифметических действий с выражениями (α) используются в виде

```
MAT B=A+(alpha) MAT B=A+(alpha)
MAT B=A/(alpha) MAT B=A*(alpha)
```

В результате их выполнения получается матрица B, все элементы которой — преобразованные элементы матрицы A. Арифметическое выражение (α) должно обязательно заключаться в скобки, матрицы A и B должны быть одинаковой размерности, допустимо совпадение имен матриц. Например, возможна запись MAT C=C+(alpha) и т. д. Матричный оператор присваивания MAT B=A создает матрицу B, элементы которой $b_{ij}=a_{ij}$. Размерности массивов для матриц A и B должны быть одинаковыми.

Операторы арифметических операций над матрицами используются в виде

```
MAT C=A+B (сложение матриц)
MAT C=A-B (вычитание матриц)
MAT C=A/B (деление матриц)
MAT C=A**B (элементы  $a_{ij}$  возводятся в степень  $b_{ij}$ )
```

Массивы матриц A, B и C должны иметь одинаковые размерности. Допускается совпадение имен массивов.

Оператор умножения матрицы на константу (для версии бейсик-плюс) имеет вид MAT B=A*K, где K — константа. Его выполнение создает матрицу B, все элементы которой $b_{ij}=a_{ij}K$.

Особенности применения матричных операторов. Применение матричных операторов существенно упрощает решение задач, связанных с использованием матриц (например, решение систем линейных уравнений). Поскольку при этом необходимы арифметические операции проводится на машинном языке, существенно сокращается время вычислений. Перед использованием матричных операторов должны быть заданы и определены (с помощью оператора DIM) массивы под исходную вычисляемую и служебные матрицы. Массивы могут иметь любые имена (а не только A, B, C, как в приведенных условных формах записи операторов). Условия совпадения имен оговорены выше.

Операторы обращения по меткам. В некоторых версиях бейсика для ускорения вычислений вводятся операторы ON# Метка и GOSUB# Метка. Метка может быть числом ($0 \div 9$) или любой латинской буквой (от A до Z). При выполнении этих операторов происходит безусловный переход (оператор ON) или обращение к подпрограмме (оператор GOSUB), которые следуют за строкой с меткой в виде REM# Метка.

Оператор-переключатель строк имеет вид ON (α) HC1, HC2, HC3. ...

где α — арифметическое выражение, указывающее порядок безусловных переходов к строкам с номерами HC1, HC2, HC3 и т. д. Если α принимает значение 1, происходит переход к строке с номером HC1, если α принимает значение 2, переход идет к строке HC2, и т. д. (номера строк не совпадают со значениями α).

Операторы ввода символьных последовательностей RED. Оператор присваивания REDL A='Символы'

присваивает переменной A символьное значение. Для такого присвоения при вводе символов с пульта используется оператор RED A. Здесь A — любая переменная.

Вывод символьной переменной, введенной операторами REDL и REDI, выполняется с помощью оператора

PRINT QA, QB, ...

Перед каждой символьной переменной из списка оператора PRINT должен стоять знак Q.

Вывод символьных переменных выполняется также операторами

```
REDP , или пробел
REDP ; или /
REDP # M [...]
REDP A [(I,K)]
```

При пустом списке оператор REDP игнорируется, точка, запятая и пробел используются в качестве разделителей, знак ; обеспечивает пропуск одной позиции, знак / — перевод строки с возвратом каретки, # M — пропуск M позиций (где M — число), A — вывод строки, введенной операторами REDL и REDI.

Широко распространенные операторы обработки символьных переменных LEN и MID описаны в версии бейсика ПЭВМ FX-702P.

Отметим, что обычно символьные переменные отождествляются с элементами одномерных и двумерных массивов, поэтому на их длину (число символов) накладываются определенные ограничения (они зависят от конкретной версии языка бейсик). В некоторых версиях бейсика имеются ограничения и на виды используемых символов (например, некоторые символы нельзя использовать в составе символьных переменных).

Все описанные выше директивы, операторы и функции присущи наиболее сложным версиям языка бейсик. В целом между отдельными версиями имеются существенные отличия, но они не затрагивают основные (характерные именно для этого языка) операторы и функции. Программы, описанные в данном справочнике, используют, по возможности, только такие операторы и функции. Пакет программ, в которых используются операторы матричных функций, дан в § П5.12.

§ 2.3. Арифметические и алгебраические операции, работа в режиме калькулятора

Арифметические и алгебраические операции микроЭВМ, программируемые на бейсике, могут выполняться как по программе, так и в режиме калькулятора. В последнем случае они проводятся по командам, вводимым без указания номера строк в режиме вычисления.

Примеры такого рода вычислений даны в табл. 2.11 (микроЭВМ FX-702P). У некоторых микроЭВМ, например Электроника-ДЗ-28, эти операции выполняются после ввода оператора PRINT (печатать). При этом аргумент функции должен заключаться в скобки.

Из приведенных примеров видно, что вычисления упрощаются при учете приоритета операций. В сомнительных случаях следует вводить скобки. При записи произведений между сомножителями обязателен знак умножения *.

В отличие от вычислений в обычных калькуляторах [10, 20], ПЭВМ обеспечивают в калькуляторном режиме выполнение не только элементарных (сложение двух чисел, вычисление одной функции и т. д.), но и сложных последовательностей действий, в частности вычисление численных значений формульных выражений. При этом в ходе ввода формулы возможна коррекция ошибочно введенных знаков, операторов и числовых констант. По существу это означает программирование разового процесса вычислений. Введенное формульное выражение в калькуляторном режиме стирается после однократного вычисления его численного значения. Отметим некоторые дополнительные виды вычислений, проводимые с использованием программ, записанных в ПЗУ ПЭВМ (т. е. микропрограмм).

Микропрограммно вычисляются и все элементарные функции, включенные в набор команд ПЭВМ.

Примеры вычислений в режиме калькулятора (для ПЭВМ FX-702P)

Операция	Показания дисплея при вводе	Результат (после нажатия клавиши EXE)
$a \leftarrow 2 \quad b \leftarrow 3$ $c \leftarrow 5 \quad d \leftarrow 1.5$	A=2 B=3 C=5 D=1.5	Переменным a, b, c и d присвоены значения 2, 3, 5 и 1,5
$2+3 \cdot 5/1,5$	$2+3 \cdot 5/1,5$	12
$a+b \cdot c/d$	$A+B \cdot C/D$	12
$5 \sin 30^\circ$	$5 \cdot \text{SIN } 30$	2,5
$\sqrt{a} \cdot \text{tg } 45^\circ$	$(\text{SQRA}) \cdot \text{TAN } 45$	1,414213562
$20 \lg (3/1,5)$	$20 \cdot \text{LOG } (3/1,5)$	6,020599913
$\frac{a+b}{c} \text{sh } d$	$((A+B)/C) \cdot \text{HSN } D$	2,129279455
$\cos 40^\circ 30' 25,2''$	$\text{COS DEG } (40,30,25,2)$	0,7603266149
$ae^d + b$	$A \cdot \text{EXP } D + B$	11,96337814
$b \cdot c!$	$B \cdot C!$	360
$2,5^{75}$	$2,5 \uparrow 75$	7,00649232E 29
$2\pi c$	$2 \cdot \pi \cdot C$	31,41592654
$(c-a) \arctg d$	$(C-A) \cdot \text{ATN } D$	168,9297974
$-2 \cdot 10^{-5} \cdot e^{-18}$	$-2E-5 \cdot \text{EXP } -18$	-3,045995949E-13
$b \ln a-c $	$B \cdot \text{LN ABS } (A-C)$	3,295836866
$(\text{int } \pi)/2$	$\text{INT } \pi/2$	1,5
$\text{int } (\pi/2) + c$	$\text{INT } (\pi/2) + C$	6

Преобразование полярных координат в прямоугольные (рис. 2.1, а) выполняется по формулам

$$r = \sqrt{x^2 + y^2} \quad \text{и} \quad \theta = \arctg (y/x),$$

где r — длина радиуса вектора. Для этого используются команды

RPC x, y EXE X (получаем r) DMS Y EXE (получаем θ)

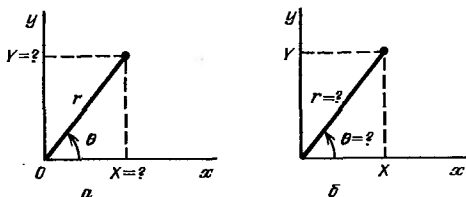


Рис. 2.1. Преобразование координат

Пример. Найти r и θ при $x = -1$ и $y = -2$.

Вводимые команды	Результат
RPC -1, -2 EXE	—
X EXE	2.236067977 = r
DMS Y EXE	$-116^\circ 33' 54.18'' = \theta$
Y EXE	$-116.5650512 = \theta$

Оператор DMS вводится, если угол θ нужно получить в градусах, минутах и секундах с десятичными долями. Угол обычно определяется в пределах $\pm 180^\circ$.

Преобразование прямоугольных координат в полярные (рис. 2.1, б) производится по формулам $x = r \cos \theta$ и $y = r \sin \theta$ с применением команд

PRC $r, \text{DEG}(\theta^\circ, \theta', \theta'')$ EXE X EXE (получаем x) Y EXE (получаем y)

Если угол вводится в градусах с десятичными долями, вместо оператора $\text{DEG}(\theta^\circ, \theta', \theta'')$ вводится значение θ .

Пример. Найти x и y при $r = 1$ и $\theta = 30^\circ$.

Вводимые команды	Результат
PRC 1, 30 EXE	—
X EXE	0.8660254038 = x
Y EXE	0,5 = y

При представлении углов в разных единицах измерения используются следующие режимы работы: MODE4 — угол в градусах, MODE5 — угол в радианах и MODE6 — угол в градах (100 град соответствует 90°).

Статистические вычисления могут выполняться в режиме калькулятора (см. подробнее § 5.3), где приведены и соответствующие формулы.

операции исполняются от начала до конца — см. пример вычисления площади круга по формуле $S = \pi D^2/4$. Словесный алгоритм при этом следующий.

1. Введем численное значение D и присвоим его переменной D ($D \leftarrow D$).
2. Вычислим $\pi D^2/4$ и присвоим полученное значение переменной S ($S \leftarrow \pi D^2/4$).
3. Выведем на печать значение переменной S .
4. Организуем останов ЭВМ.

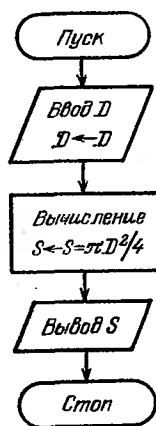


Рис. 2.3. Алгоритм вычисления площади круга

Соответствующий графический алгоритм показан на рис. 2.3, а, программа имеет вид

```

05 REM 'ВЫЧИСЛЕНИЕ ПЛОЩАДИ КРУГА'
10 INPUT 'ВВЕДИТЕ ДИАМЕТР КРУГА D=' D
20 LETS=#PI*D^2/4
30 PRINT 'ПЛОЩАДЬ КРУГА S=' S
40 END
  
```

Обратите внимание на комментарии при операторах INPUT в строке 10 и PRINT в строке 30.

У разветвляющихся программ вычисления производятся в различных частях в зависимости от заданных исходных данных или результатов вычислений. Это обеспечивают операторы условных переходов. На рис. 2.4

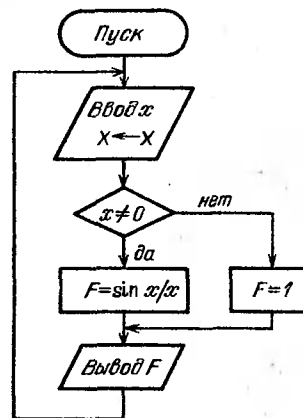


Рис. 2.4. Алгоритм вычисления функции $\sin x/x$

показан графический алгоритм вычисления функции $F(x) = \sin x/x$ при $x \neq 0$ и $F(x) = 1$ при $x = 0$. Соответствующая программа приведена выше (в § 2.1) при описании операторов IF...THEN.

Циклические программы обеспечивают циклическое (т. е. повторяющееся) выполнение отдельных фрагментов заданное или конечное, но неопределенное число раз — до получения результата с заданной погрешностью. Циклы с заданным числом повторений организуются с помощью операторов FOR ... TO ... STEP и NEXT. Примеры правильной и неправильной организации ряда циклов даны на рис. 2.5.

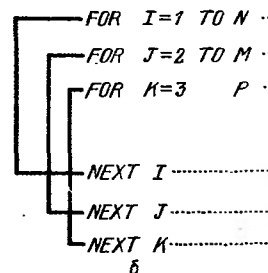
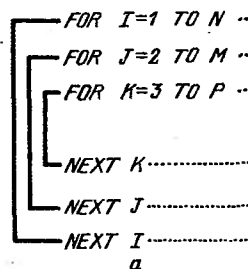


Рис. 2.5. Правильная (а) и неправильная (б) организации циклов с помощью операторов FOR и NEXT

Пример. Построение циклической программы с циклом, повторяющимся заданное число раз. Пусть надо вычислить N -е число Фибоначчи. Напомним, что числа Фибоначчи образуют последовательность, у которой каждый очередной член равен сумме двух предыдущих:

$$0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8 \quad 13 \quad 21 \quad 34 \quad \dots$$

$$\qquad \qquad \qquad \underbrace{\qquad \qquad \qquad}_{2+3=5} \qquad \qquad \underbrace{\qquad \qquad \qquad}_{13+21=34}$$

Словесное описание алгоритма может быть следующим.

1. Задаем число N .
2. Присвоим переменной A значение 0 ($A \leftarrow 0$), а переменной B значение 1 ($B \leftarrow 1$).
3. Организуем цикл вычислений $C = A + B$, организовав счетчик цикла с помощью переменной I , значение которой должно меняться от начального $I = 3$ до конечного $I = N$ с шагом, равным 1. В конце каждого числа проведем замену переменных: $A \leftarrow B$ и $B \leftarrow C$.

4. При $I=N$ задан выход из цикла и выведем на индикацию число C .

5. Перейдем к выполнению п. 1 с помощью операции безусловного перехода.

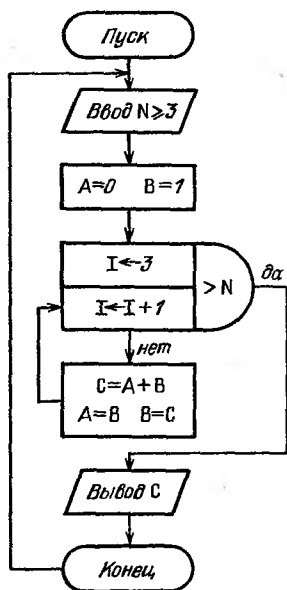


Рис. 2.6. Алгоритм вычисления чисел Фибоначчи

Графический алгоритм решения этой задачи показан на рис. 2.6. Программа имеет вид

```
10 INPUT 'ЗАДАЙТЕ N'>=3 N=N:LETA=0:LETB=1
20 FOR I=3 TO N STEP 1
30 LETC=A+B:LETA=B:LETB=C
40 NEXT I
50 PRINT 'FN='C:GOTO 10:END
60 PRINT!F1.9!'FN='C
70 GOTO 20:END
```

В строке 10 организован ввод значения N и присвоение переменным A и B значений 0 и 1. В строке 20 задан заголовок цикла и указаны пределы изменения (от 3 до N) управляющей переменной I . Далее в строке 30 проводится вычисление переменной $C = A + B$, присвоение переменной A значения B и переменной B значения C . В строке 40 задан возврат из цикла, если $I=N$. И, наконец, в строке 50 задается печать значения переменной C и безусловный переход к строке 10. Задав, например, $N=10$, получим результат 34.

Пример. Построение циклической программы с циклом, повторяющимся до получения результата с заданной точностью. Пусть надо вычислить обратный гиперболический тангенс $\text{arth } x$ с заданной точностью $\varepsilon = E$, используя разложение в ряд

$$\text{arth } x = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots, \quad x < 1.$$

Для этого будем вычислять сумму членов x^i/i для $i \leftarrow i + 2$ при начальном $i=1$.

Графический алгоритм вычисления показан на рис. 2.7.

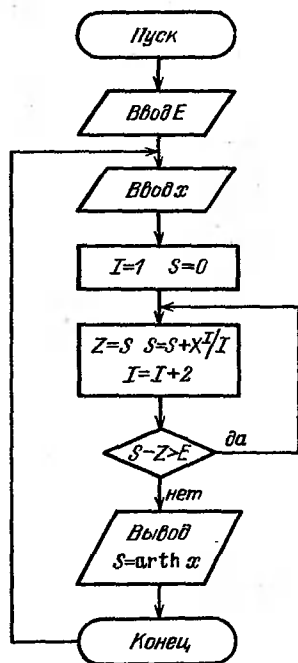


Рис. 2.7. Алгоритм вычисления функции $\text{arth } x$ разложением ее в ряд

В программе (см. с. 46, сверху) цикл организован в строках 30 и 40 с помощью операторов `IF ... THEN`, ведущих сравнение разности $(S-Z)$, где Z — предшествующее значение S , с числом E , задающим погрешность. Если $(S-Z) > E$, то организуется условный переход к выполнению строки 30. В противном случае выдается на индикацию или печать результат вычислений (строка 50) и происходит безусловный переход (строка 60) к строке 20, т. е. ввод нового значения аргумента x , и вычислению $\text{arth } x$.

Подобным образом строится большинство программ вычисления специальных функций по их разложениям в ряд, вычисления рядов и сложных числовых последовательностей. Следует отметить, что остановка счета в данном случае возможна, если ряд является сходящимся.

Обращения к подпрограммам производятся в том случае, когда отдельные фрагменты программ должны выполняться при различных значениях исходных данных для их вычисления данных. Подпрограммы, как и циклы, могут вкладываться друг в друга (см. рис. 2.8). При завершении выполнения подпрограммы происходит возврат в исходную программу и выполнение оператора, следующего за обращением `GOSUB n` к подпрограмме, начинающейся со строки n . Это возвращение задается оператором `RETURN` в конце подпрограммы.

Пример. Построение программы с подпрограммой. Пусть нужно вычислить

```

10 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E=' E
20 INPUT 'ВВЕДИТЕ X=' X: LET I=1: LET S=0
30 LET Z=S: LET S=S+X*I/1: LET I=I+2
40 IF S-Z>E THEN 30
50 PRINT 'АНТ(X)=' S
60 GOTO 20: END
    
```

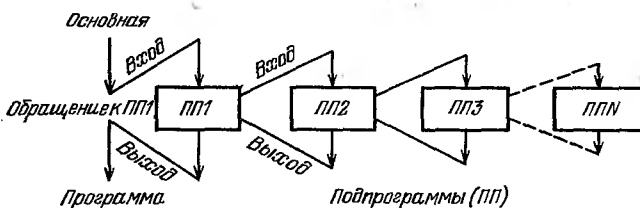


Рис. 2.8. Структура организации подпрограмм с вложениями их друг в друга

определенный интеграл

$$I = \int_a^b \sqrt{2x+1} dx = \int_a^b f(x) dx$$

по простой формуле Симпсона

$$I \approx \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{6}\right) + f(b) \right]. \quad (2.1)$$

В этом случае нужно трижды вычислить значение подынтегральной функции $f(x) = \sqrt{2x+1}$ для $x=a$, $(a+b)/2$ и b . Поэтому вычисление $f(x)$ целесообразно вынести в подпрограмму.

```

10 INPUT 'ВВЕДИТЕ ПРЕДЕЛЫ ИНТЕГРИРОВАНИЯ A,B ' A,B
20 LET X=A: GOSUB 60: LET S=F
30 LET X=(A+B)/2: GOSUB 60: LET S=S+4*F
40 LET X=B: GOSUB 60: LET S=S+F
50 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' S*(B-A)/6: STOP
55 REM 'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(X)'
60 LET F=SQR(2*X+1): RETURN
70 END
    
```

Здесь подпрограмма с оператором возврата занимает строку 60. Суммирование членов в квадратных скобках (2.1) производится с помощью вспомогательной переменной S. Для $a=0$ и $b=1$ вычисления по этой программе дают результат $I=1,398150843$.

Оценка времени вычислений производится суммированием времен выполнения отдельных операций с учетом повторяемости их в циклах и подпрограммах. Для приближенной оценки времени выполнения отдельных операций можно использовать следующую методику. Вначале определяют время проведения N циклов ($N=100$ или 1000). Для этого, задав N , пускают простейшую программу

```

10 FOR I=1 TO N: NEXT I
20 PRINT 'END': END
    
```

и засекают время ее исполнения. Деля его на N , получают время исполнения пустого цикла $t_{ц}$. Затем внутрь цикла вставляется команда с заданной операцией, например $LET C=A+B$, если нужно найти время проведения операции сложения. Снова определяют общее время вычислений и делят его на N . От полученного времени отнимают $t_{ц}$ и получают время исполнения заданной операции $t_{оп}$.

Время $t_{оп}$ сильно зависит от типа ПЭВМ, поэтому подобные испытания следует провести для конкретной ПЭВМ, имеющейся в распоряжении пользователя (разумеется, если нужных данных нет в ее описании). Ниже приведены типовые значения $t_{оп}$ для СПП

на базе настольной ЭВМ Электроника-Д3-28 и ПЭВМ класса Pocket Computers FX-702P ($t_{оп}$ в секундах):

Операция	$t_{оп}$ для ПЭВМ	
	Электроника-Д3-28	FX-702P
Сложение и вычитание	0,03	0,05
Умножение и деление	0,04	0,065
Функция \sqrt{x}	0,06	0,07
Функции $\ln x$, $\lg x$, e^x	0,05	0,18
Тригонометрические функции	0,1	0,25
Обратные тригонометрические функции	0,05—0,1	0,3
Гиперболические функции	0,065	0,2

Операция	$t_{оп}$ для ПЭВМ	
	Электроника-ДЗ-28	FX-702P
Обратные гиперболические функции	0,1	0,3
Функции $\ln(x)$, $\text{abs}(x)$	0,025	0,04
Функция y^x	0,08	0,33
Безусловный переход	0,01	0,03
Условный переход	0,04	0,06
Обращение к подпрограмме и выход из нее	0,02	0,04
Один цикл (t_u)	0,012	0,02

В данном случае арифметические операции сложения, вычитания, умножения и деления выполняются почти за одинаковое время. Это не общее правило. У многих ПЭВМ операция деления производится медленнее, чем операция умножения, а последняя, в свою очередь, выполняется медленнее, чем операции сложения и вычитания. Самой медленной обычно является операция возведения в степень (y^x).

Сложность подсчета числа операций, особенно если число циклов в программе не фиксировано, а задается погрешностью вычислений, делает целесообразным определение общего времени выполнения вычислений по данным решения типового контрольного примера. Исключения возможны, если время выполнения всей программы менее 5—10 с.

Оптимизация программ проводится с целью сокращения времени их исполнения и объема памяти ПЭВМ, придания программам компактности и алгоритмической ясности. Поскольку эти качества часто взаимопротиворечивы, то процесс оптимизации носит субъективный характер.

Для большинства пользователей ПЭВМ вполне справедливо правило — составляйте программу так, как это вам нравится и как это вам удобно. Это правило справедливо, если вычисления по программе носят единственный или эпизодический характер. В этом случае большие затраты времени на тщательную оптимизацию программы нерациональны. Тем более, что нередко программу, составленную за десяток минут, можно оптимизировать многие дни, хотя конечный результат остается не же. Большинство начинающих пользователей, как правило, размещают в каждой строке один оператор. Выше отмечались недостатки таких программ: чрезмерная длина листинга программы, неэкономное использование ОЗУ, трудность обозрения сразу всей программы, большой расход бумаги при распечатке программ принтером. Поэтому лучше сразу привыкнуть к размещению в строке нескольких простых операторов, если к иному не обязывает алгоритм вычислений. Программы, рассчитанные на многократное использование, желательно тщательно оптимизировать.

Несмотря на субъективный характер подготовки программ, есть ряд приемов, которые следует использовать при составлении и оптимизации программ. Прежде всего необходимо особо внимательно отнестись к выбору алгоритма. Порой затраты на это в 10—20 мин в дальнейшем сокращает время массовых вычислений на многие часы, сутки и даже месяцы. Например, переход от обычного спектрального анализа к процедуре быстрого преобразования Фурье (БПФ) часто ведет к уменьшению времени счета в десятки и даже сотни раз.

В программах надо учитывать реальное время выполнения операций (см. выше). Например, если ПЭВМ выполняет деление медленнее, чем умножение, то операцию вида $B = A/2$ лучше записать как $B = .5 * A$. Возведение в малую целую степень y^x (при $x = 2, 3, \dots$) лучше заменить умножением, например $Y \rightarrow 2$ (или $Y \uparrow 2$) следует заменить операцией $Y * Y$. При табуляции функций иногда полезно использовать не полностью завершенные циклы следующей конструкции:

```
FOR I=A TO IE99 STEP H:...:NEXT I
```

Цикл практически не завершается, так как конечное значение управляющей переменной I задано очень большим (10^{89}). Такой цикл аналогичен цепочке операторов

```
HCI LET I=A
```

```
HCM LET I=I+H
```

```
HCN GOTO HCI
```

но выполняется существенно быстрее (выигрыш во времени до 3—5 раз при пустом цикле, т. е. учете затрат времени только на выполнение операторов в строках HCM и HCN). Выход из цикла, если он нужен, задается рабочими операторами, записанными вместо многоточия.

Арифметические выражения в программах целесообразно приводить к виду, позволяющему уменьшить число операций. Например, вычисление значения $c = a^2 - 2ab + b^2$ при четной записи $c = a \cdot a + 2 \cdot a \cdot b - b \cdot b$ требует четырех операций умножения и по одной операции сложения и вычитания. Представив $c = a - b$ и $c = c * c$, т. е. учитывая соотношение $(a^2 - 2ab + b^2) = (a - b)^2$, получим тот же результат всего при одной операции вычитания и одной умножения. Аналогичным образом вычисление $y = e^a / e^b$ лучше выполнить в виде $y = e^{a-b}$, что уменьшит число обращений к микропрограммам вычисления функций. В простых программах такие преобразования опускаются, если они идут в ущерб наглядности программы или приводят к отказу от общепринятой формы записи вычисляемого арифметического выражения. Эффективным приемом упрощения вычислительных операций и придания им более общего вида является нормирование арифметических выражений.

Следует по возможности убирать повторяющиеся фрагменты вычислений и сосредоточивать их в одном месте. Особое внимание необходимо уделять «чистке циклов», т. е.

выносу за пределы циклов повторяющихся фрагментов вычислений. Например, выражение

$$S = \sum_{i=1}^{30} \sum_{j=1}^{100} \frac{\ln a_i e^{ix}}{i+j}$$

требует вычисления $\ln a_i$ и e^{ix} 3000 раз. Между тем, представив это выражение в виде

$$S = \sum_{i=1}^{30} \left(\ln a_i \sum_{j=1}^{100} \frac{e^{ix}}{i+j} \right),$$

можно сразу сократить число вычислений $\ln a_i$ до 30. Если позволяет память ЭВМ, можно отдельно сформировать массив экспонент $E(j) = e^{ix}$ и сократить до 100 число вычислений e^{ix} . Однако при этом будет иметь место обращение к массиву $E(j)$ 3000 раз. Тем не менее последний вариант по затратам машинного времени будет лучшим.

При использовании переменных из массивов нередко повторяются их индексы. Например, при каждом использовании выражения

$$A(J * K + 1) = A(J * K + 1) + R$$

к переменной $A(J * K + 1)$ прибавляется значение R . Однако индекс $(J * K + 1)$ вычисляется при этом дважды — в левой и в правой частях равенства. Замена этого выражения фрагментом программы

$$I = J * K + 1; A(I) = A(I) + R$$

ведет к экономии памяти (выражение стало короче) и сокращению времени счета (индекс $I = J * K + 1$ вычисляется один раз). Эта экономия будет еще существеннее, если переменные $A(J * K + 1)$ используются в программе более двух раз, например в циклических фрагментах.

Часто полезно заменять индексированную переменную простой переменной. Например, вычисление арифметического выражения

$$Y = A(I * K + 1) * A(I * K + 1) + X$$

можно выполнить в виде

$$Z = A(I * K + 1); Y = Z * Z + X$$

Сокращение времени вычислений в этом случае связано и с тем, что ЭВМ опознает простую переменную быстрее, чем индексированную.

Иногда, в какой-то мере копируя подход, принятый при составлении программ на языке фортран, программы составляются по блочному принципу: делается основная программа, в которой обеспечивается обращение к вспомогательным подпрограммам. Если последние снабжены комментариями, в целом программа оказывается очень наглядной. Однако в общем случае не следует злоупотреблять обращениями к подпрограммам, так как на них затрачивается дополнительное время (особенно, когда обращения идут из циклов).

Необходимо также отметить, что использование особых «программных трюков» в настоящее время считается недостатком, а не достоинством программ. Сокращение времени вычислений не является самоцелью. В правильно составленных программах принимаются меры по устранению неоправданных потерь машинного времени. Поэтому рекомендации по составлению программ не следует абсолютизировать.

В некоторых версиях языка бейсик имеются специальные операторы для включения в программы фрагментов, выполняемых на других языках, например на машинно-ориентированном языке конкретной ПЭВМ. Последнее позволяет уменьшить время выполнения программ (иногда в десятки раз), сохранить присущий бейсику диалог ПЭВМ с пользователем и заметно расширить круг решаемых на данной ПЭВМ задач [29]. Однако этот прием носит частный характер, при его применении теряется наглядность самих программ, усложняется их отладка и исключается применение программ для ПЭВМ с другим машинно-ориентированным языком.

§ 2.5. Специальные вопросы программирования на языке бейсик

Описанные в § 2.4 приемы программирования направлены на решение простых вычислительных задач. Рассмотрим некоторые специальные вопросы программирования, выявляющие более полно возможности языка бейсик.

Организация диалога с пользователем.

Простейший диалог с пользователем заложен в самой программе-интерпретаторе. Так, у большинства ПЭВМ загрузка этой программы (с ПЗУ или магнитной ленты) сопровождается начальным диалогом: подтверждается загрузка интерпретатора, у пользователя запрашивается, с каким оборудованием он будет работать, будут ли использованы внешние подпрограммы и т. д. (см. Приложения 1 и 3).

Комментарии могут вводиться пользователем в составе операторов INPUT (при вводе), PRINT (при выводе результатов вычислений) и REM (внутри программы). Характер комментариев всецело задается пользователем. Он может быть очень кратким, например

```
10 INPUT D
```

(при этом индицируется лишь знак ?), или детальным:

```
10 INPUT 'ВВЕДИТЕ ЧИСЛЕННОЕ
      ЗНАЧЕНИЕ ДИАМЕТРА
      ОКРУЖНОСТИ D='D
```

Следует помнить, что слишком подробные комментарии удлинит программу, увеличат время счета, загружат память ЭВМ, затруднят чтение программ. Такие комментарии оправданы лишь в отдельных случаях —

когда программа рассчитывала, используя оператор TAB, который не понимает сути вычислений. Специалист может довольствоваться короткими комментариями, например

```
10 INPUT 'D='D
```

(если он знает, что речь идет о площади круга, то вряд ли нужно особо указывать, что D — это диаметр окружности).

Индикация ошибок предусмотрена у большинства ПЭВМ. Сложные ПЭВМ дают подробное указание об ошибке, например такое:

```
NET ЗАКРЫВАЮЩЕЙ СКОБКИ В
ОПЕРАТОРЕ LET В СТРОКЕ 150
```

Более простые ПЭВМ дают указания вида ОШИБКА <Номер> В СТРОКЕ <Номер>

При этом характер ошибок определяется по их номеру с помощью специальных таблиц (см. Приложения 2 и 4).

Вывод простых графиков. Многие ПЭВМ способны формировать сложные многоцветные графические изображения. Однако даже те ПЭВМ, у которых нет специальных команд построения графиков, могут создавать простые графические изображения с помощью операторов PRINT и TAB.

Рисунки с помощью оператора PRINT формируются с помощью знаков, помещенных в апострофы или кавычки. Так, из знаков — можно создавать штриховые горизонтальные прямые, из знаков ! — вертикальные линии и т. д. Ниже дана программа построения стилизованной электрической схемы. Она строится из различных знаков, присущих ПЭВМ без графики.

Схема, сформированная в поле комментариев PRINT, будет выведена на экран дисплея или печать после исполнения оператора RUN.

Оператор TAB может использоваться для графического отображения решений; построения графиков функций, колебательных процессов и т. д. При этом функция $Y(X)$ строится с осью Y поперек, а с осью X — вдоль направления печати. Зона печати при использовании оператора TAB разбивается на $N_{\text{макс}}$ ($N_{\text{макс}}$ — обычно 80 или 100) позиций, каждая из которых задается целой частью арифметического выражения. Так, приведенная ниже программа обеспечивает построение графика функции $Y=X^2$, который строится с помощью знаков *.

В этой программе изменение аргумента X от значения -6 до $+6$ задается циклом с началом в строке 10 и концом в строке 60. Если $X \neq 0$, осуществляется печать значений $Y=X^2$ с помощью операторов PRINT TAB в строке 35 знаками *. Одновременно печатается знак !, образующий горизонтальную ось графика. При $X=5$ и $X=-5$ (см. строки 40 и 50) вместо знака ! печатаются численные значения $X=5!$ и $X=-5!$ на оси X , т. е. осуществляется масштабирование оси X . При $X=0$ вместо печати значения $Y=X^2=0$ в строке 25 задана печать оси Y знаками — и цифрами 0, 10, 20 и 30, обеспечивающими ее масштабирование. График функции $Y=X^2$, формируемый этой программой, дан на рис. 2.9.

У ПЭВМ, рассчитанных на вывод графической информации, имеются специальные операторы для построения графиков. Так, оператор PLOT X, Y, Z обеспечивает вывод точки с координатами X и Y (используются целые части значений переменных X и Y). Если $Z=0$, точка погашена, если $Z=1$, она высвечивается. С помощью оператора PLOT можно по точкам строить сложные графики, например, если Y и X связаны соотношением $Y=X^2$, то будет построена линия, подобная приведенной на рис. 2.9 (однако график при

```
01 PRINT'ПРИМЕР ПОСТРОЕНИЯ ЭЛЕКТРОННОЙ СХЕМЫ'
02 PRINT'
03 PRINT'
04 PRINT'
05 PRINT'
06 PRINT'
07 PRINT'
08 PRINT'
09 PRINT'
10 PRINT'
11 PRINT'
12 PRINT'
13 PRINT'
20 END
```

```
05 PRINT#1'ПРИМЕР ПОСТРОЕНИЯ ГРАФИКА ФУНКЦИИ Y=X^2'
10 FOR X=-6 TO 6 STEP .5
20 IF X<>0 THEN 35
25 PRINT'0-----10-----20-----30---Y=X^2'
30 GO TO 60
35 PRINT'!'TAB(X^2)'*'
40 IF X=-5 THEN PRINT'X=-5';
50 IF X=5 THEN PRINT'X=5';
60 NEXT X:END
```

Пример построения графика функции $y=x^2$

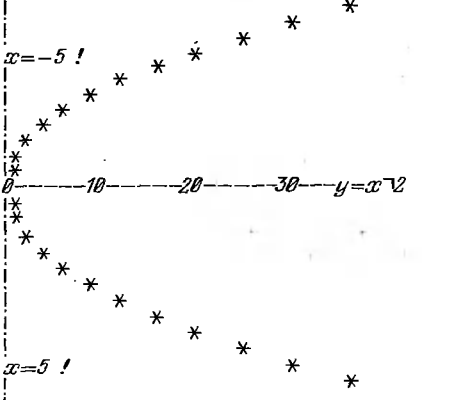


Рис.2.9. График функции $Y=X^2$ с координатными осями

этом может иметь более высокую разрешающую способность). С помощью оператора LINE X, Y вычерчивается прямая линия с концом, заданным целыми значениями X и Y, и началом, заданным предшествующим оператором PLOT. Например, операторы PLOT 0, 0, 1 и LINE 50, 50 обеспечивают построение прямой с началом в точке (0,0) и концом в точке (50,50) декартовой системы координат на плоскости. С помощью этих операторов можно строить динамические, т. е. движущиеся изображения и сложные рисунки.

Некоторые типы принтеров (например, Электроника-УВВПЧ-30-004, входящие в комплект последних моделей СПП на базе микроЭВМ Электроника-Д3-28) могут формировать высококачественные одноцветные графические изображения по точкам, причем печать графики возможна как вдоль, так и поперек бумажной ленты.

Диалог с применением кодов. Часто нужно на ПЭВМ выполнить ту или иную операцию, например, провести расчет по той или иной формуле. Это можно сделать, используя операторы INPUT и IF, THEN.

```

10 INPUT 'ВВЕДИТЕ X=';X
20 PRINT 'ДЛЯ ВЫЧИСЛЕНИЯ X^2 УКАЖИТЕ КОД 2'
30 INPUT 'ДЛЯ ВЫЧИСЛЕНИЯ X^3 УКАЖИТЕ КОД 3 КОД=';K
40 IF K=2 THEN 60
50 IF K=3 GOTO 70: GOTO 20
60 PRINT 'X^2=';X^2: GOTO 10
70 PRINT 'X^3=';X^3: GOTO 10:END
    
```

В этой программе при задании по запросу ЭВМ кода 2 вычисляется значение x^2 , а при задании кода 3 — x^3 .

При ошибочно введенном коде повторяется запрос кода.

```

10 INP "X",X
20 A$=KEY$:IF A$=" " THEN 20
30 IF A$="2" THEN 200
40 IF A$="3" THEN 300
50 GOTO 20
200 PRT"X^2="; X^2
300 PRT"X^3="; X^3: GOTO 10: END
    
```

Диалог с применением клавиш пользователя. У ряда ПЭВМ есть команды присвоения клавишам функций клавиш пользователя. При этом пуск программы с заданного места производится нажатием только клавиш пользователя. Ниже дана программа для ПЭВМ FX-702P, иллюстрирующая применение клавиш пользователя.

После ввода значения x символьной переменной A\$ командой KEY придается (строка 20) свойство управляющей переменной. Теперь, если нажать клавишу 2, оператор IF в строке 30 передаст управление строке 200; если нажать клавишу 3, аналогичный оператор в строке 40 передаст управление строке 300. Во всех других случаях происходит переход к строке 20 и ПЭВМ ждет нажатия одной из клавиш пользователя. Если ввести $x=2$ и нажать клавишу 2, будет получен результат $X \uparrow 2=4$, если нажать клавишу 3, то $X \uparrow 3=8$ (напомним, что в системе команд FX-702P стрелка \uparrow означает возведение в степень).

Диалог с применением символьных констант. Он наиболее близок к естественному, так как при нем машина задает вопросы, а пользователь отвечает на них на естественном языке (словесном или математическом). Иллюстрацией к организации такого диалога на ПЭВМ FX-702P является следующая программа:

```

10 INP "X",X
20 INP "X^2 OR X^3 ?";$
30 IF $="X^2" THEN 200
40 IF $="X^3" THEN 300
50 GOTO 20
200 PRT"X^2="; X^2: GOTO 10
300 PRT"X^3="; X^3: GOTO 10
    
```

Здесь после задания x ПЭВМ в строке 20 задает пользователю вопрос: $X \uparrow 2$ OR $X \uparrow 3$? (x^2 или x^3). Надо ответить фразой $X \uparrow 2$ или $X \uparrow 3$. Если ответ был $X \uparrow 2$, символьной переменной \$ присваивается значение $X \uparrow 2$, а в строке 30 операторы IF ... THEN передают управление строке 200, т. е. вычислению и выдаче значения $X \uparrow 2=x^2$. Если задать $X \uparrow 3$, то в строке 40 операторы IF ... THEN передадут управление строке 300, т. е. вычис-

лению и выдаче значения $X \uparrow 3$. Во всех других случаях безусловный переход GOTO 20 в строке 50 ведет к повторению запроса ПЭВМ.

Переходы с косвенной адресацией. Они возможны в расширенных версиях бейсика. Так, у ПЭВМ FX-702P возможны 6 типов таких переходов:

1. GOTO γ
2. GOTO $\neq \gamma$
3. IF α <условие> β : GOTO γ
4. IF α <условие> β : GOTO $\neq \gamma$

7. GSB \neq γ

6. GSB \neq γ

Здесь α , β и γ — переменные или арифметические выражения. Номер строки или программной области (при знаке \neq перед γ) задается целой частью арифметического выражения или значения переменной γ . Это значение не должно выходить за пределы 9 при адресации к программным областям и 10 000 при адресации к строкам γ .

Применение косвенной адресации поясним таким примером: пусть надо выполнить интегрирование нескольких функций по формуле Симпсона, не меняя основную программу. Для этого можно каждую функцию обозначить номером N ($N=1, 2, 3 \dots$) и вычислять подпрограммой, записанной, например, со строк 100N (100 при $N=1$, 200 при $N=2$ и т. д.). Соответствующая программа имеет вид

```
05 INF "F N=",N
10 INF "A",A,"B",B
20 X=A: GSB N*100: S=F
30 X=(A+B)/2: GSB N*100: S=S+4*XF
40 X=B: GSB N*100: S=S+F
50 PRT "I=": S*(B-A)/6: STOP
100 F=SCR(2*X+1): RET
200 F=1/(1+X^2): RET
300 F=10*LN X: RET
500 END
```

При задании $N=1, 2$ или 3 эта программа будет вычислять соответственно интегралы

$$1) \int_a^b \sqrt{2x+1} dx; \quad 2) \int_a^b \frac{dx}{1+x^2};$$

$$3) \int_a^b 10 \ln x dx.$$

Задав, например, $N=1$, $a=0$ и $b=1$, получим для первого интеграла значение $I=1,398150843$.

§ 2.6. Перевод программ с одной версии бейсика на другую

Различия между версиями бейсика могут заключаться в синтаксисе (правилах записи операторов) и в словарном запасе операторов.

Различия в синтаксисе, как правило, носят непринципиальный характер и сводятся к замене апострофов на кавычки, двоеточия на точку с запятой, знаков возведения в степень (\rightarrow на \uparrow) и т. д. Такая замена легко формализуется, но требует знания синтаксиса обеих версий бейсика — той, с которой переводят, и той, на которую переводят программы.

Различия в словарном запасе операторов могут быть весьма существенными. Так, простые версии бейсика содержат до 20—50 операторов, а сложные — до 200—300. Эти различия нередко носят принципиальный характер. Так, операторы косвенной адресации, работы с целочисленными и символъ-

ными переменными и константами, вывода графиков и др., применяемые в сложных версиях бейсика, не имеют функциональных аналогов в простых версиях. Поэтому если перевод с простой версии на сложную достаточно прост и формален, то обратный перевод весьма сложен, а подчас и принципиально невозможен.

Перевод программ справочника на другие версии бейсика достаточно прост благодаря специальным мерам, принятым при составлении программ. Так, направленность справочника на научно-технические, статистические и экономические расчеты позволила отказаться от использования операторов вывода сложных графиков и работы с символическими (текстовыми) переменными. Все программы справочника даны на версии бейсика умеренной сложности, применяемой в массовых системах подготовки программ на базе настольной микроЭВМ Электроника-ДЗ-28. В эту версию бейсика входит основной набор операторов (табл. 2.1—2.4), применяемых в других версиях бейсика (для микроЭВМ Электроника-С60, Искра-226, диалоговых вычислительных комплексов ДВК-1, ДВК-2 и др.). Принятая версия бейсика является расширенной в части операторов вычисления математических функций (табл. 2.6), что упрощает запись программ. Однако перевод операторов этой версии на язык других версий (где эти операторы отсутствуют) носит формальный характер (см. табл. 2.12) и поэтому достаточно прост. Для облегчения перевода в прикладных программах, как правило, не используются операторы DATA, READ, RESTORE и ON, поскольку они легко заменяются другими (INPUT, LET, GOTO) и отсутствуют у некоторых версий бейсика (например, ПЭВМ FX-702P).

Общие рекомендации по переводу программ. При переводе программ рекомендуется придерживаться следующих правил.

1. Внимательно изучить особенности версий бейсика (оригинала и той, которая нужна пользователю). В ходе этого изучения нужно выяснить, насколько родственны эти версии, прежде всего по словарному запасу операторов.

2. Если версии бейсика достаточно близки, следует использовать построчный или пооператорный перевод, учитывая при этом различие в синтаксисе этих версий.

3. Если в оригинале встречаются операторы, отсутствующие в нужной версии, следует выяснить их смысл и заменить другими операторами. Так, для замены ряда операторов вычисления математических функций можно использовать табл. 2.9.

4. Если версии бейсика сильно различаются, целесообразно составить детальный описательный или графический алгоритм решения нужной задачи и затем по нему заново составить программу на нужной версии бейсика.

Некоторые практические рекомендации по переводу программ. При переводе пользователь может существенно изменить характер

Формальные правила перевода операторов вычисления математических функций для некоторых версий бейсика (P=π=3,141592654)

Версии бейсика

СПП на базе микроЭВМ Электроника-ДЗ-28	МикроЭВМ Электроника-С60, Искра-226, комплексы ДВК-1, ДВК-2	ПЭВМ FX-702P
DEG(X)	180*X/P	MODE4
RAD(X)	P*X/180	MODE5
TAN(X)	SIN(X)/COS(X)	TAN X
ASN(X)	ATN(X/SQR(1-X*X))	ASN X
ACS(X)	P/2-ATN(X/SQR(1-X*X))	ACS X
HSN(X)	(EXP(X)-EXP(-X))/2	HSN X
HCS(X)	(EXP(X)+EXP(-X))/2	HCS X
HTN(X)	(EXP(X)-EXP(-X))/(EXP(X)+EXP(-X))	HTN X
AHS(X)	LOG(X+SQR(X*X+1))	AHS X
AHC(X)	LOG(X+SQR(X*X-1))	AHC X
AHT(X)	LOG((1+X)/(1-X))/2	AHT X
LGT(X)	LOG(X)/LOG(10)	LOG X
LOG(X)	LOG(X)	LN X
X-INT(X)	X-INT(X)	FRAC X
EXT(X)	X↑10	X↑10

диалога с ПЭВМ (сократить или, напротив, расширить его). При переводе зарубежных программ целесообразно перевести комментарии с английского языка на русский. У пользователей зарубежными ПЭВМ может возникнуть необходимость в обратном переводе.

У некоторых версий бейсика (например, для микроЭВМ Электроника-С60) комментарии не вводятся с помощью оператора INPUT. В этом случае при переводе используется оператор PRINT.

Пример.

Версия для микроЭВМ Электроника-ДЗ-28

```
10 INPUT 'ВВЕДИТЕ X, Y' X, Y
```

Версия для микроЭВМ Электроника-С60

```
10 PRINT "ВВЕДИТЕ X, Y"
15 INPUT X, Y
```

Переадресация и переиндексация массивов и переменных бывает необходима при их специальной организации, применяемой в некоторых версиях бейсика. Например, в версии бейсик-CASIO, применяемой в ПЭВМ FX-702P, все массивы (одномерные и двумерные) организуются на основе одной переменной A: например, A(I), A(J, I). В большинстве других версий переменные массивов могут обозначаться любыми буквами (или буквами с цифрой от 0 до 9): X(I), Z9(I, J) и т. д. Поэтому прямой перевод программ с версии бейсик-CASIO на другие версии возможен, а обратный требует переадресации массивов. Одновременно возникает необходимость и в их переиндексации. Например, три одномерных массива X(I), Y(J), Z(K) при $I < 10$, $J < 20$ и $K < 10$ в версии бейсик-CASIO можно заменить массивами A(I), A(10+I) и A(30+I). В этой версии $A5=A(5)$, $A(15)=B5$, тогда как в других версиях A5 и A(5) — совершенно разные

переменные, как и A(15) и B5. В связи с этим, если одна из версий бейсика при переводе программ имеет подобные особенности, к их учету следует относиться с особым вниманием.

Приведение аргумента функций в допустимые пределы нередко необходимо при переводе программ. Например, аргумент тригонометрических функций в версии бейсик-CASIO должен лежать в пределах от -8π до $+8\pi$ рад. В ряде других версий бейсика таких ограничений нет. Выход аргумента за допустимые пределы ведет к остановке вычислений с выдачей знака ошибки. Для приведения аргумента X в заданные пределы можно использовать фрагмент программы вида

```
50 IF X ≤ 8*π THEN 70
60 X=X-8*π: GOTO 50
70
```

или

```
50 X=2*π*FRAC(X/2/π)
```

В последнем случае аргумент приводится к интервалу от -2π до $+2\pi$.

Редактирование и отладка переведенных программ. Редактирование программ, полученных путем перевода, сводится к проверке правильности записи всех операторов. Если перевод сделан с простой версии бейсика на более сложную, следует изучить возможность оптимизации программ, используя специфические возможности применяемой версии бейсика. Например, выражение вида $X=\sqrt{A^2+B^2}$ на ПЭВМ FX-702P может вычисляться с помощью оператора RPC A, B вместо выражения $X=\text{SQR}(A*A+B*B)$ и т. д. Отладка полученных переводом программ существенно облегчается, если программа-оригинал была снабжена контрольными тестами (примерами). При редак-

тировании и отладке программ следует использовать заложенный в основу бейсика начальный диалог. Так, при неточностях в записи операторов и в их синтаксисе ПЭВМ подсказывает пользователю характер и место ошибок, что позволяет легче найти и устранить их.

Иногда причиной грубых расхождений результатов вычислений при правильном переводе программ являются особенности округления чисел при исполнении операторов циклов вида `FOR $\alpha = \beta$ TO γ STEP δ` . Малые погрешности в вычислении γ , например, когда γ есть целое число в целой степени, могут приводить к тому, что число циклов окажется на 1 меньше или больше, чем предполагаемое. Например, число циклов, заданных оператором `FOR I=1 TO N 73`, при $N=2$ может быть не 8, а 7 или 9 в зави-

симости от того, как округляется N^3 . У некоторых ПЭВМ округление при возведении в целую степень целых чисел специально предусмотрено. Однако у других, в частности у Электроники-ДЗ-28, этого нет (хотя при вычислении N^3 с помощью оператора `PRINT N 73` будет получен точный результат — ошибка содержится в скрытых разрядах). В подобных ситуациях округление можно проводить с помощью оператора `INT`. Например, выполнение точно N^3 циклов гарантировано, если оператор цикла записать в виде `FOR I=1 TO INT (N 73 + 1)`. Небольшая добавка 0,1 к N^3 создает определенность в операции округления. Подобные ситуации возможны и при использовании операторов косвенной адресации вида `ON α , GOTO α , GOSUB α` , когда α задает номер строки, т. е. целое число.

АЛГОРИТМЫ И ПРОГРАММЫ ЭЛЕМЕНТАРНЫХ ВЫЧИСЛЕНИЙ

§ 3.1. Операции с действительными числами

Перевод чисел из одной системы счисления в другую выполняется по правилам, описанным в [36]. Для перевода целого числа X_M с основанием M в число X_N с основанием N число X_M делится на N' ($i=1, 2, 3, \dots$), а последовательность остатков от деления умножается на M^j ($j=0, 1, 2, \dots$) и складывается. Числа X_M в виде правильных дробей последовательно умножаются на N' ($i=1, 2, 3, \dots$), целые части результата делятся на M^j ($j=1, 2, 3, \dots$). В приведенных ниже программах операции над целыми и дробными частями X_M проводятся одновременно. Количество цифр X_M и X_N не должно превышать разрядности ЭВМ.

Программа 3.1.

```

10 PRINT 'ПРЕОБРАЗОВАНИЕ ЧИСЛА XM В ЧИСЛО XN'
20 INPUT 'ВВЕДИТЕ XM=':X: INPUT 'ВВЕДИТЕ M=':M
25 INPUT 'ВВЕДИТЕ N=':N: LETU=0
30 LETS=10: LETK=10: LETM=0: GOSUB 50
35 LETS=L: LETK=N: LETX=U: LETM=10
40 GOSUB 50: PRINT !8.4! 'XN=':U: GOTO 20
50 LETL=K: LETD=1: LETE=1: LETU=0
60 LETC=INT(X): LETW=X-C
70 FOR I=S TO 0 STEP -1
80 LETB=W*L: LETA=INT(B)
90 LETD=D*M: LETU=A/D+U
100 LETT=C: LETW=W-A: IF W=0 THEN 120
110 NEXT I
120 LETA=INT(T/L)
130 LETU=(T-A*L)*E+U: LETE=E*M
140 LETT=A: IF A<>0 THEN 120
150 RETURN: END

```

Контрольные примеры. $X_8=27 \rightarrow X_5=43$ ($M=8, N=5$), $X_5=43 \rightarrow X_8=27$ ($M=5, N=8$), $X_{10}=100 \rightarrow X_2=1100100$ ($M=10, N=2$), время счета $t_c \approx 5$ с.

Нахождение наибольшего общего делителя двух чисел N и M проводится с помощью рекуррентного соотношения

$$N_k = N_{k-2} - \text{Int}(N_{k-2}/N_{k-1}) N_{k-1}, \\ k=2, 3, \dots,$$

где $N_0 = \max(|N|, |M|)$, $N_1 = \min(|N|, |M|)$. Если N_k станет нулевым, то наибольший общий делитель $S = (N, M) = N_{k-1}$. Числа N и M должны быть меньше 10^9 для ЭВМ с разрядностью 10.

Программа 3.2.

Пример. Для $N=23345$, $M=9135$ получим $(N, M)=1015$, а для $N=238$ и $M=347$ $(N, M)=34$.

Разложение целого числа M на простые множители производится последовательным делением M на 2 и ряд простых чисел d ($d=3, 5, 7, 9, 11, 13, \dots$) по формуле

$$M_i = M_{i-1}/d$$

до тех пор, пока соблюдается условие $\sqrt{M_i} + 1 \leq d$.

Программа 3.3.

Пример. Число 1234567890 раскладывается на множители $2 \times 3 \times 3 \times 5 \times 3607 \times 3803$.

Преобразование десятичного числа X в цепную, или непрерывную, дробь производится выделением целой части X , а затем

целых частей $X_i = 1/(X_{i-1} - \text{int } X_{i-1})$, где $\text{int } X_{i-1}$ — целая часть X_{i-1} .

Программа 3.4.

Пример. Число $X = \pi = 3,14159264 \rightarrow [3, 7, 15, 1, 252, 1, 1, 1, 4]$, т. е.

$$\pi = 3 + \frac{1}{7 + \frac{1}{15 + \frac{1}{1 + \frac{1}{252 + \dots}}}}$$

Преобразование цепной дроби в десятичное число X производится непосред-

Программа 3.2.

```
10 PRINT 'НАХОЖДЕНИЕ НАИБОЛЬШЕГО ОБЩЕГО ДЕЛИТЕЛЯ'  
15 INPUT 'ВВЕДИТЕ ПЕРВОЕ ЧИСЛО N=' N  
20 INPUT 'ВВЕДИТЕ ВТОРОЕ ЧИСЛО M=' M  
30 LET I=INT(N/M):LET L=M  
40 LET M=N-M*I:LET N=L  
50 IF M<>0 THEN 30  
60 PRINT!9.0!'НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ=' N  
70 GOTO 15:END
```

Программа 3.3.

```
05 PRINT 'РАЗЛОЖЕНИЕ ЧИСЛА X НА ПРОСТЫЕ МНОЖИТЕЛИ'  
10 INPUT 'ВВЕДИТЕ X=' A  
15 PRINT!9.0!'МНОЖИТЕЛИ X':GOTO 40  
20 PRINT 2:LETA=A/2  
30 IF A-1=0 THEN 50  
40 IF A/2-INT(A/2)=0 THEN 20  
50 LET B=3  
60 LET C=SQR(A)+1  
70 IF B>=C THEN 140  
80 IF A/B-INT(A/B)=0 THEN 100  
90 LET B=B+2:GOTO 70  
100 IF A/B*B-A=0 THEN 120  
110 GOTO 90  
120 PRINT B  
130 LET A=A/B:GOTO 60  
140 PRINT A:PRINT 'КОНЕЦ':GOTO 10:END
```

Программа 3.4.

```
05 PRINT 'ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО ЧИСЛА В ЦЕПНУЮ ДРОБЬ'  
10 INPUT 'ВВЕДИТЕ ДЕСЯТИЧНОЕ ЧИСЛО X=' X  
15 FOR I=1 TO 10  
20 LET F=INT(X):PRINT!9.0!F  
30 LET X=1/(X-F):NEXT I  
40 GOTO 10:END
```

Программа 3.5.

```
05 PRINT 'ПРЕОБРАЗОВАНИЕ ЦЕПНОЙ ДРОБИ В ДЕСЯТИЧНОЕ ЧИСЛО'  
10 INPUT C:LETD=1:LETA=0:LETF=C:LETB=1  
20 PRINT F/B:INPUT E:LETD=C  
30 LETC=C*E+D:LETD=0:LETD=B  
40 LETB=B*E+A:LETA=0:LETF=C  
50 GOTO 20:END
```

Программа 3.6.

```
10 PRINT 'ДЕЛЕНИЕ ЧИСЛА A НА ЧИСЛО B'  
20 PRINT 'С ПРОИЗВОЛЬНОЙ ТОЧНОСТЬЮ'  
30 INPUT 'ВВЕДИТЕ A,B ' A,B  
40 LET I=INT(A/B):PRINT!9.0!'ЦЕЛАЯ ЧАСТЬ РЕЗУЛЬТАТА ' I  
50 PRINT 'ПОСЛЕДОВАТЕЛЬНОСТЬ ЦИФР РЕЗУЛЬТАТА'  
60 LET F=INT(A/B):PRINT!1.0! F;  
70 LET A=(A-F*B)*10:GOTO 60:END
```

ственный вычислением X по формуле цепной дроби.

Программа 3.5.

Пример. [3, 7, 15, 1, 292, 1, 1, ..., 4] → → 3,141592654.

Деление числа A на число B с произвольной точностью выполняется выделением целой части отношения A/B при вычислении $F = \text{int}(A/B)$ и затем использованием рекуррентной формулы $A_i = (A_{i-1} - F \times B) \times 10$.

Программа 3.6.

Пример. 32/9,3 = 3,4408602150537 ... Цифры результата выдаются до остановки ЭВМ пользователем.

Получение N -го числа Фибоначчи, т. е. числа из последовательности

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

у которой новый член равен сумме двух предыдущих ($C_1=0$; $C_2=1$):

$C_i = C_{i-1} + C_{i-2}$, $i=3, 4, 5, \dots$

Программа 3.7.

```
10 PRINT 'ВЫДАЧА N-ГО ЧИСЛА ФИБОНАЧЧИ'
20 LET A=0: LET B=1: INPUT 'ВВЕДИТЕ N>=3 N=' N
30 FOR I=3 TO N
40 LET C=A+B: LET A=B: LET B=C
50 NEXT I
60 PRINT I;F1.9 I;FN='C
70 GOTO 20:END
```

Для $N=10$ получим $C_{10}=34$.

§ 3.2. Операции и функции с комплексными числами и переменными

Комплексные числа записываются в алгебраической форме $Z = A + jB = \text{Re} Z + j \text{Im} Z$ или в тригонометрической $Z = \text{Re}^{j\theta}$, причем $R = \sqrt{A^2 + B^2}$ и $\theta = \text{arctg}(B/A)$.

Сложение и вычитание комплексных чисел $Z_1 = A_1 + jB_1$ и $Z_2 = A_2 + jB_2$ выполняется по формулам

$$Z_1 + Z_2 = (A_1 + A_2) + j(B_1 + B_2),$$

$$Z_1 - Z_2 = (A_1 - A_2) + j(B_1 - B_2).$$

Программа 3.8.

```
05 PRINT I;F1.9!
10 PRINT 'АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ С КОМПЛЕКСНЫМИ'
20 PRINT ' ЧИСЛАМИ В АЛГЕБРАИЧЕСКОЙ ФОРМЕ'
30 INPUT 'ВВЕДИТЕ RE Z1, IM Z1 'A,B
40 INPUT 'ВВЕДИТЕ RE Z2, IM Z2 'C,D
50 INPUT 'ВВЕДИТЕ КОД ОПЕРАЦИИ +1,-2,*3,/4 'K
60 IF K=1 THEN 110
70 IF K=2 THEN 130
80 IF K=3 THEN 140
90 IF K=4 THEN 160
100 PRINT 'КОД УКАЗАН НЕВЕРНО':GOTO 50
110 LET A=A+C:LET B=B+D
120 PRINT 'Z 0='A' + J*(B)':GOTO 40
130 LET A=A-C:LET B=B-D:GOTO 120
140 LET E=A*C-B*D:LET F=A*D+C*B
150 LET A=E:LET B=F:GOTO 120
160 LET L=C*D:LET E=A*C+B*D:LET F=B*C-A*D
170 LET A=E/L:LET B=F/L:GOTO 120:END
```

Умножение и деление комплексных чисел $Z_1 = A_1 + jB_1$ и $Z_2 = A_2 + jB_2$ выполняется по формулам

$$Z_1 Z_2 = (A_1 A_2 - B_1 B_2) + j(A_1 B_2 + B_1 A_2),$$

$$\frac{Z_1}{Z_2} = \frac{A_1 A_2 + B_1 B_2}{A_2^2 + B_2^2} + j \frac{B_1 A_2 - A_1 B_2}{A_2^2 + B_2^2}, \quad (3.1)$$

а в тригонометрической форме

$$Z_1 Z_2 = \text{Re}^{j\theta} = R \cos \theta + j R \sin \theta,$$

где $R = R_1 R_2$, $\theta = \theta_1 + \theta_2$, и

$$Z_1 / Z_2 = \text{Re}^{j\theta} = R \cos \theta + j R \sin \theta,$$

где $R = R_1 / R_2$, $\theta = \theta_1 - \theta_2$.

Программа 3.8.

Контрольный пример. Вычисленные выражения

$$\frac{(5-j \cdot 3)(3+j \cdot 2)}{(5+j \cdot 3)(2-j \cdot 4)} + (0,5+j \cdot 1)$$

дает результат $\text{Re} Z_0 = 1,158823529$ и $\text{Im} Z_0 = 1,464705882$, т. е. $Z_0 = 1.158823529 + j \times 1,464705882$.

Программа 3.9.

Для $Z_1 = 2e^{j \cdot 0,5}$ и $Z_2 = 3e^{j \cdot 0,25}$ получим $Z_1 Z_2 = Z_0 = 6e^{j \cdot 0,75}$ и $Z_1 / Z_2 = Z_0 = 1,5e^{j \cdot 0,25}$

Алгебраические функции комплексного переменного:

$$Z^2 = (A^2 - B^2) + j2AB,$$

$$\frac{1}{Z} = \frac{A}{A^2 + B^2} - j \frac{B}{A^2 + B^2},$$

$$\frac{1}{Z^2} = \frac{A^2 - B^2}{(A^2 + B^2)^2} - j \frac{2AB}{(A^2 + B^2)^2},$$

$$\sqrt{Z} = \pm \sqrt{\frac{A + \sqrt{A^2 + B^2}}{2}} \pm j \sqrt{\frac{-A + \sqrt{A^2 + B^2}}{2}},$$

$$\exp Z = e^Z = e^A \cos B + j e^A \sin B,$$

$$\ln Z = \frac{1}{2} \ln (A^2 + B^2) + j (\text{arctg} \frac{B}{A} \pm 2\pi k).$$

$k=0, 1, 2, \dots$

```

10 PRINT 'УМНОЖЕНИЕ И ДЕЛЕНИЕ КОМПЛЕКСНЫХ '
20 PRINT 'ЧИСЕЛ В ТРИГОНОМЕТРИЧЕСКОЙ ФОРМЕ'
30 INPUT 'В КАКОЙ ФОРМЕ Z1 0-АЛГЕБ. 1-ТРИГОНОМ. ' K
40 IF K=1 THEN 70
50 INPUT 'ВВЕДИТЕ RE Z1, IM Z1 ' A,B
60 LET R1=SQR(A*A+B*B):LET Q1=ATN(B/A):GOTO 80
70 INPUT 'ВВЕДИТЕ R1, Q1 ' R1,Q1
80 INPUT 'В КАКОЙ ФОРМЕ Z2 0-АЛГЕБ. 1-ТРИГОНОМ. ' K
90 IF K=1 THEN 120
100 INPUT 'ВВЕДИТЕ RE Z2, IM Z2 ' C,D
110 LET R2=SQR(C*C+D*D):LET Q2=ATN(D/C):GOTO 130
120 INPUT 'ВВЕДИТЕ R2, Q2 ' R2,Q2
130 INPUT 'ВВЕДИТЕ КОД ОПЕРАЦИИ 0 *, 1 / ' K
140 IF K=0 THEN 160
150 LET R=R1/R2:LET Q=Q1-Q2:GOTO 170
160 LET R=R1*R2:LET Q=Q1+Q2
170 PRINT 'Z0=' R '* EXP(J*( ' Q '))'
180 LET R1=R:LET Q1=Q:GOTO 80:END

```

$$\sqrt[n]{Z} = \sqrt[n]{R} e^{jA_k} = \sqrt[n]{R} \cos A_k + j \sqrt[n]{R} \sin A_k, \quad A_k = \theta/n + 360k/n, \quad k = 0, 1, 2, \dots, n-1.$$

$$Z^n = R^n e^{jn\theta} = R^n \cos n\theta + j R^n \sin n\theta.$$

Программа 3.10.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=Z^2'
20 INPUT 'ВВЕДИТЕ RE Z=' A: INPUT 'ВВЕДИТЕ IM Z=' B
30 LET E=A^2-B^2:LET F=2*A*B
40 LET A=E:LET B=F
50 PRINT 'RE F(Z)=' A: PRINT 'IM F(Z)=' B
60 GOTO 20:END

```

Пример. $(3+j-2)^2 = 5+j-12$.

Программа 3.11.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=1/Z'
20 INPUT 'ВВЕДИТЕ RE Z=' A: INPUT 'ВВЕДИТЕ IM Z=' B
30 LET G=A^2+B^2:LET E=A/G
40 LET F=-B/G:LET A=E:LET B=F
50 PRINT 'RE F(Z)=' A: PRINT 'IM F(Z)=' B
60 GOTO 20:END

```

Пример. $1/(3+j-2) = 0.2307692308 - j \cdot 0.1538461538$.

Программа 3.12.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=1/Z^2'
20 INPUT 'ВВЕДИТЕ RE Z=' A: INPUT 'ВВЕДИТЕ IM Z=' B
30 LET G=(A^2+B^2)^2:LET E=(A^2-B^2)/G
40 LET F=-2*A*B/G:LET A=E:LET B=F
50 PRINT 'RE F(Z)=' A: PRINT 'IM F(Z)=' B
60 GOTO 20:END

```

Пример. $1/(3+j-2)^2 = 0.02958579882 - j \cdot 0.07100591716$.

Программа 3.13.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=SQR(Z)'
20 INPUT 'ВВЕДИТЕ RE Z=' A: INPUT 'ВВЕДИТЕ IM Z=' B
30 LET G=SQR(A^2+B^2):LET E=SQR((A+G)/2)
40 LET F=SQR((-A+G)/2):LET A=E:LET B=F
50 PRINT 'RE Z=+' A: PRINT 'IM Z=+' B
60 GOTO 20:END

```

Пример. $\sqrt{3+j-2} = \pm 1.81735402 + j \cdot 0.5502505227$.

Программа 3.14.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=EXP(X)'
20 INPUT 'ВВЕДИТЕ RE Z=' A: INPUT 'ВВЕДИТЕ IM Z=' B
30 LET E=EXP(A)*COS(B):LET F=EXP(A)*SIN(B)
40 LET A=E:LET B=F
50 PRINT 'RE EXP(Z)=' A: PRINT 'IM EXP(Z)=' B
60 GOTO 20:END

```

Пример. $\exp(3+j-2) = -8.358532651 + j \cdot 18.26372704$.

Программа 3.15.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=LOG(Z)'
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B
30 LET E=LOG(A^2+B^2)/2: LET F=ATN(B/A)
40 LET A=E: LET B=F
50 PRINT 'F='A'+J*( 'B' +2*K*#PI)'
60 GOTO 20:END

```

Пример. $\ln(3+j\cdot 2) = 1,282474679 + j(0,5880026035 + 2k\pi)$.

Программа 3.16.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ F(Z)=Z^(1/N)'
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B
30 INPUT 'ВВЕДИТЕ N='N: LET X=SQR(A^2+B^2)
35 LET Y=ATN(B/A): LET A=X^(1/N)*COS(Y/N)
40 LET B=X^(1/N)*SIN(Y/N): PRINT 'RE F(Z)='A
50 PRINT 'IM F(Z)='B:END

```

Пример. $\sqrt[4]{3+j\cdot 2} = 1,36311829 + j\cdot 0,2018352064$.

Программа 3.17.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ F(Z)=Z^N'
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B
30 INPUT 'ВВЕДИТЕ N='N: LET X=SQR(A^2+B^2)
40 LET Y=ATN(B/A): LET A=X^N*COS(Y*N)
50 LET B=X^N*SIN(Y*N): PRINT 'RE F(Z)='A
60 PRINT 'IM F(Z)='B:END

```

Пример. $(3+j\cdot 2)^3 = -8,999999999 + j\cdot 46$.

Тригонометрические функции комплексного переменного:

$$\sin Z = \sin A \operatorname{ch} B + j \cos A \operatorname{sh} B,$$

$$\cos Z = \cos A \operatorname{ch} B - j \sin A \operatorname{sh} B,$$

$$\operatorname{tg} Z = \frac{\sin 2A}{\cos 2A + \operatorname{ch} 2B} + j \frac{\operatorname{sh} 2B}{\cos 2A + \operatorname{ch} 2B}.$$

Программа 3.18.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=SIN(Z)'
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B
30 LET E=SIN(A)*HCS(B): LET F=COS(A)*HSN(B)
40 LET A=E: LET B=F
50 PRINT 'RE F(Z)='A: PRINT 'IM F(Z)='B
60 GOTO 20:END

```

Пример. $\sin(3+j\cdot 2) = 0,5309210863 - j\cdot 3,59056459$.

Программа 3.19.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=COS(Z)'
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B
30 LET E=COS(A)*HCS(B): LET F=-SIN(A)*HSN(B)
40 LET A=E: LET B=F
50 PRINT 'RE F(Z)='A: PRINT 'IM F(Z)='B
60 GOTO 20:END

```

Пример. $\cos(3+j\cdot 2) = -3,724545505 - j\cdot 0,51182257$.

Программа 3.20.

```

10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=TAN(Z)'
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B
30 LET A=A^2: LET B=B^2: LET G=COS(A)+HCS(B)
35 LET E=SIN(A)/G: LET F=HSN(B)/G
40 LET A=E: LET B=F
50 PRINT 'RE F(Z)='A: PRINT 'IM F(Z)='B
60 GOTO 20:END

```

Пример. $\operatorname{tg}(3+j\cdot 2) = -0,009884375039 + j\cdot 0,965385879$.

Обратные тригонометрические функции комплексного переменного:

$$\arcsin Z = \arcsin B_k + j \ln [A_k + \sqrt{A_k^2 - 1}],$$

$$\arccos Z = \arccos B_k - j \ln [A_k + \sqrt{A_k^2 - 1}],$$

$$\operatorname{arctg} Z = \frac{1}{2} \operatorname{arctg} \left(\frac{2A}{1 - A^2 - B^2} \right) + \frac{j}{4} \ln \left(\frac{A^2 + (B + j)^2}{A^2 - (B - j)^2} \right),$$

где $A_k = \frac{1}{2} \sqrt{(A+1)^2 + B^2} + \frac{1}{2} \sqrt{(A-1)^2 + B^2}$ и $B_k = \frac{1}{2} \sqrt{(A+1)^2 + B^2} - \frac{1}{2} \sqrt{(A-1)^2 + B^2}$.

Программа 3.21.

```
10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=ASN(Z)'  
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B  
25 LETC=SQR((A+1)^2+B^2)/2  
30 LETD=SQR((A-1)^2+B^2)/2  
35 LETE=C+D: LETF=C-D: LETA=ASN(F)  
40 LETB=LOG(E+SQR(E*E-1))  
50 PRINT 'RE F(Z)='A: PRINT 'IM F(Z)='B  
60 GOTO 20: END
```

Пример. $\arcsin(3+j \cdot 2) = 0,9646585044 + j \cdot 1,968637926$.

Программа 3.22.

```
10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=ACS(Z)'  
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B  
25 LETC=SQR((A+1)^2+B^2)/2  
30 LETD=SQR((A-1)^2+B^2)/2  
35 LETE=C+D: LETF=C-D: LETA=ACS(F)  
40 LETB=-LOG(E+SQR(E*E-1))  
50 PRINT 'RE F(Z)='A: PRINT 'IM F(Z)='B  
60 GOTO 20: END
```

Пример. $\arccos(3+j \cdot 2) = 0,6061378224 - j \cdot 1,968637926$.

Программа 3.23.

```
10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=ANT(Z)'  
20 INPUT 'ВВЕДИТЕ RE Z='B: INPUT 'ВВЕДИТЕ IM Z='A: LETB=-B  
25 LETC=SQR((A+1)^2+B*B)/2  
30 LETE=ATN(2*A/(1-A*A-B*B))/2  
35 LETF=LOG((A*A+(B+1)^2)/(A*A+(B-1)^2))/4  
40 LETB=E: LETA=-F  
50 PRINT 'RE F(Z)='A: PRINT 'IM(Z)='B  
60 GOTO 20: END
```

Пример. $\operatorname{arctg}(0,3+j \cdot 0,2) = 0,3018746667 + j \cdot 0,1849946201$.

Гиперболические функции комплексного переменного:

$$\operatorname{sh} Z = \operatorname{sh} A \cos B + j \operatorname{ch} A \sin B,$$

$$\operatorname{ch} Z = \operatorname{ch} A \cos B + j \operatorname{sh} A \sin B,$$

$$\operatorname{tg} Z = \frac{\operatorname{sh} A}{\operatorname{ch} 2A + \cos 2B} + j \frac{\sin 2B}{\operatorname{ch} 2A + \cos 2B}.$$

Программа 3.24.

```
10 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=HSN(Z)'  
20 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B  
30 LETE=HSN(A)*COS(B): LETF=HCS(A)*SIN(B)  
40 LETA=E: LETB=F  
50 PRINT 'RE F(Z)='A: PRINT 'IM F(Z)='B  
60 GOTO 20: END
```

Пример. $\operatorname{sh}(3+j \cdot 2) = -4,16890696 + j \cdot 9,154499147$.

Программа 3.25.

```

10 PRINT'ВЫЧИСЛЕНИЕ F(Z)=HCS(Z)'
20 INPUT'ВВЕДИТЕ RE Z='A:INPUT'ВВЕДИТЕ IM Z='B
30 LETE=HCS(A)*COS(B):LETF=HNS(A)*SIN(B)
40 LETA=E:LETB=F
50 PRINT'RE F(Z)='A:PRINT'IM F(Z)='B
60 GOTO 20:END

```

Пример. $\text{ch}(3+j \cdot 2) = -4.189625691 + j \cdot 9.109227894$.

Программа 3.26.

```

10 PRINT'ВЫЧИСЛЕНИЕ F(Z)=HTN(Z)'
20 INPUT'ВВЕДИТЕ RE Z='A:INPUT'ВВЕДИТЕ IM Z='B
30 LETA=A*2:LETB=B*2:LETG=HCS(A)+COS(B)
35 LETE=HNS(A)/G:LETF=SIN(B)/G
40 LETA=E:LETB=F
50 PRINT'RE F(Z)='A:PRINT'IM F(Z)='B
60 GOTO 20:END

```

Пример. $\text{th}(3+j \cdot 2) = 1.003238627 - j \cdot 0.0037664025642$.

Обратные гиперболические функции комплексного переменного:

$\text{arsh } Z = -j \arcsin(jZ)$, $\text{arch } Z = \pm j \arccos Z$, $\text{arth } Z = -j \text{arctg}(jZ)$.

Программа 3.27.

```

10 PRINT'ВЫЧИСЛЕНИЕ F(Z)=AHS(Z)'
20 INPUT'ВВЕДИТЕ RE Z='B:INPUT'ВВЕДИТЕ IM Z='A:LETB=-B
25 LETC=SQR((A+1)^2+B*B)/2
30 LETD=SQR((A-1)^2+B*B)/2
35 LETE=C+D:LETF=C-D:LETB=ASN(F)
40 LETA=LOG(E+SQR(E*E-1))
50 PRINT'RE F(Z)='A:PRINT'IM F(Z)='B
60 GOTO 20:END

```

Пример. $\text{arsh}(3+j \cdot 2) = 1.98338703 + j \cdot 0.5076527843$.

Программа 3.28.

```

10 PRINT'ВЫЧИСЛЕНИЕ F(Z)=AHS(Z)'
20 INPUT'ВВЕДИТЕ RE Z='A:INPUT'ВВЕДИТЕ IM Z='B
25 LETC=SQR((A+1)^2+B*B)/2
30 LETD=SQR((A-1)^2+B*B)/2
35 LETE=C+D:LETF=C-D:LETB=ACS(F)
40 LETA=LOG(E+SQR(E*E-1))
50 PRINT'RE F(Z)='+-'A:PRINT'IM F(Z)='+-'B
60 GOTO 20:END

```

Пример. $\text{arch}(3+j \cdot 2) = \pm 1.968637926 \pm j \cdot 0.6061378224$.

Программа 3.29.

```

10 PRINT'ВЫЧИСЛЕНИЕ F(Z)=ATH(Z)'
20 INPUT'ВВЕДИТЕ RE Z='B:INPUT'ВВЕДИТЕ IM Z='A:LETB=-B
25 LETC=SQR((A+1)^2+B*B)/2
30 LETE=ATH(2*A/(1-A*A-B*B))/2
35 LETF=LOG((A*A+(B+1)^2)/(A*A+(B-1)^2))/4
40 LETB=E:LETA=-F
50 PRINT'RE F(Z)='A:PRINT'IM(Z)='B
60 GOTO 20:END

```

Пример. $\text{arth}(0,3+j \cdot 0,2) = 0,2957499203 + j \cdot 0,2154744937$.

В описанных программах значения $\text{Re } F(Z)$ и $\text{Im } F(Z)$ придают переменным A и B ($A \leftarrow \text{Re } F$, $B \leftarrow \text{Im } F$). Поэтому, опустив комментарии, программы можно использовать в качестве подпрограмм вычисления функций комплексного переменного.

§ 3.3. Вычисление степенных многочленов и дробно-рациональных функций

Схема Горнера применяется для вычисления степенных многочленов $P(Z)$ (полиномов),

$$P(Z) = a_n Z^n + a_{n-1} Z^{n-1} + \dots + a_1 Z + a_0,$$

и имеет вид

$$P(Z) = (\dots((a_n Z + a_{n-1}) Z + a_{n-2}) Z + \dots + a_1) Z + a_0.$$

Коэффициенты $a_n, a_{n-1}, \dots, a_1, a_0$ будем считать действительными. При действительном аргументе $x=Z$ схема Горнера применяется непосредственно, при комплексном $Z=A+jB$ — с применением операций умножения и сложения комплексных чисел (см. § 3.2).

Программа 3.30.

```
10 PRINT 'ВЫЧИСЛЕНИЕ СТЕПЕННОГО МНОГОЧЛЕНА'
15 INPUT 'ВВЕДИТЕ СТЕПЕНЬ МНОГОЧЛЕНА N=':N:DIM A(N)
20 FOR I=N TO 0 STEP -1
30 PRINT!2.0!'ВВЕДИТЕ КОЭФФИЦИЕНТ A(I)'
35 INPUT A(I):NEXT I
40 INPUT 'ВВЕДИТЕ ЗНАЧЕНИЕ X=':X:LET Y=0
50 FOR I=N TO 1 STEP -1
60 LET Y=(Y+A(I))*X:NEXT I
70 LET Y=Y+A(0):PRINT!F1.9!'ЗНАЧЕНИЕ P(X)=' Y
80 GOTO 40:END
```

Для многочлена $P(x) = 4x^4 + 3x^3 + 2x^2 + 1x + 0$ получим $P(1) = 10, P(2) = 98$.

Программа 3.31.

```
10 PRINT 'ВЫЧИСЛЕНИЕ ПОЛИНОМА С КОМПЛЕКСНЫМ АРГУМЕНТОМ'
15 INPUT 'ВВЕДИТЕ СТЕПЕНЬ ПОЛИНОМА N=':N:DIM A(N)
20 FOR I=N TO 0 STEP -1
30 PRINT!2.0!'ВВЕДИТЕ КОЭФФИЦИЕНТ A(I)='?'
35 INPUT 'A(I):NEXT I
40 INPUT 'ВВЕДИТЕ RE(Z)=':X:INPUT 'ВВЕДИТЕ IM(Z)=':Y
45 LET R=A(N):LET J=0
50 FOR I=1 TO N
60 LET E=X*R-Y*J+A(N-I)
70 LET F=Y*R+X*J:LETR=E:LETJ=F
80 NEXT I:PRINT!F1.9!'
90 PRINT 'RE P(X)=':R:PRINT 'IM P(X)=':J
100 GOTO 40:END
```

Для многочлена $P(Z) = 8Z^8 + 7Z^7 + 6Z^6 + \dots + 1Z + 1$ получим $P(1+j \cdot 1) = 144 - j \cdot 113$.

Программа 3.32.

```
10 PRINT 'ОЦЕНКА УСТОЙЧИВОСТИ МНОГОЧЛЕНА'
20 INPUT 'ЗАДАЙТЕ СТЕПЕНЬ МНОГОЧЛЕНА N=':N:DIM A(N)
30 FOR I=N TO 0 STEP -1:PRINT!2.0!'A(I)='
40 INPUT A(I):NEXT I:LET W=N-2
50 LET N=W+2:LET Z=N:LET B=A(N):LET N=N-1:LET C=A(N):LET N=N-2
60 IF C=0 GOTO 140
70 LET D=B*(ABS(N)):LET N=N+1
80 LET A(ABS(N))=(C*(A(ABS(N))-D))/C
90 IF A(ABS(N))<0 THEN 140
100 LET V=3-N:IF V>0 THEN 120
110 LET N=V:GOTO 70
120 LET W=Z-3:IF W<0 THEN 50
130 PRINT 'МНОГОЧЛЕН УСТОЙЧИВ':STOP
140 PRINT 'МНОГОЧЛЕН НЕУСТОЙЧИВ':END
```

Устойчивость многочлена, описывающего линейную систему с обратной связью (например, электронный усилитель), оценивается с помощью критерия Гурвица—Гурвица. Для устойчивости необходимо, чтобы значение модуля многочлена $A(Z) = A(p)$, где $p = j\omega$, при фазовом угле $\theta = 0$ было менее 1 (т. е. годограф системы при ω , меняющемся от 0 до ∞ , не должен охватывать точку 0,1 комплексной плоскости). Более удобно судить об устойчивости $A(p)$ по его разложению в цепную дробь $[c_1 p, c_2 p, \dots, c_n p]$. Если все коэффициенты c_i этого разложения положительны, многочлен устойчив. Оценка устойчивости многочлена

$$A(p) = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0$$

выполняется с помощью следующей программы.

Пример. Многочлен $A(p) = 2p^4 + 3p^3 + p^2 + 5p + 1$ неустойчив, а многочлен $A(p) = 12p^5 + 12p^4 + 16p^3 + 10p^2 + 5p + 1$ устойчив.

Вычисление значений дробно-рациональной функции комплексного переменного

$$F(Z) = \frac{A(Z)}{B(Z)} = \frac{a_m Z^m + a_{m-1} Z^{m-1} + \dots + a_1 Z + a_0}{b_n Z^n + b_{n-1} Z^{n-1} + \dots + b_1 Z + b_0}$$

производится вычислением комплексных значений многочленов $A(Z)$ и $B(Z)$ с последующим делением их по формуле (3.1).

Программа 3.33.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ДРОБНО-РАЦИОНАЛЬНОЙ ФУНКЦИИ'
20 PRINT 'F(Z)=A(Z)/B(Z) КОМПЛЕКСНОГО ПЕРЕМЕННОГО'
30 INPUT 'ВВЕДИТЕ СТЕПЕНИ МНОГОЧЛЕНОВ A(Z) И B(Z) M,N 'M,N
40 DIM A(M),B(N):FOR I=M TO 0 STEP -1
50 PRINT!2.0!'ВВЕДИТЕ A'I:INPUT A(I):NEXT I
60 FOR I=N TO 0 STEP -1:PRINT'ВВЕДИТЕ B'I
70 INPUT B(I):NEXT I
80 INPUT'ВВЕДИТЕ RE Z, IM Z'X,Y:LETR=A(M):LETJ=0
90 FOR I=1 TO M:LETE=X*R-Y*J+A(M-I)
100 LETF=Y*R+X*J:LETR=E:LETJ=F:NEXT I
110 LETQ=R:LETU=J:LETR=B(N):LETJ=0
120 FOR I=1 TO N:LETE=X*R-Y*J+B(N-I)
130 LETF=Y*R+X*J:LETR=E:LETJ=F:NEXT I
140 LETC=R*R+F*F:LETD=Q*Q:LETR=U*U
150 LETE=U*R-Q*F:LETP=D/C:LETI=E/C
160 PRINT!F1.9!'RE F(Z)='P:PRINT 'IM F(Z)='I
170 LETM1=SQR(P*P+I*I):LETD=ATN(I/P)
175 PRINT'МОДУЛЬ M='M
180 PRINT'ФАЗА В РАДИАНАХ Q='Q
190 PRINT'ФАЗА В ГРАДУСАХ Q='Q*180/#PI
200 GOTO 80:END

```

Контрольный пример. Для функции

$$F(Z) = \frac{3Z^3 + 2Z^2 + Z + 0,5}{5Z^5 + 4Z^4 + 3Z^3 + 2Z^2 + Z + 0,2}$$

где $m=3$ и $n=5$, при $Z=3+j \cdot 2$ получим $F(3+j \cdot 2) = 0,01850739412 - j \cdot 0,040551793708$. Программа 3.33 обеспечивает также вычисление модуля

МОДУЛЬ $F(Z) = \sqrt{[\operatorname{Re} F(Z)]^2 + [\operatorname{Im} F(Z)]^2}$
и фазы
ФАЗА $F(Z) = \operatorname{arctg} [\operatorname{Im} F(Z) / \operatorname{Re} F(Z)]$

Программа 3.34.

```

05 PRINT 'ДЕЛЕНИЕ МНОГОЧЛЕНА A(X) СТЕПЕНИ N'
10 PRINT 'НА ДВУХЧЛЕН A(X)=X^2+B1*X+B0'
20 INPUT 'ВВЕДИТЕ N,B1,B0 'N,C,B:DIM A(N)
30 FOR I=N TO 0 STEP -1
40 PRINT!2.0!'ВВЕДИТЕ КОЭФФИЦИЕНТ A'I '
45 INPUT A(I):NEXT I
50 LETV=A(N):LETN=N-1:LETX=A(N):LETN=N-1
60 FOR I=N TO 0 STEP -1
80 GOSUB 100:NEXT I
90 GOTO 120
100 LETD=X-Y*C:LETE=A(I):LETA(I)=V
110 LETX=E-V*B:LETY=D:RETURN
120 PRINT!F1.9!'ОСТАТОК R(X)='Y'*X + ('X')'
130 PRINT'КОЭФФИЦИЕНТЫ МНОГОЧЛЕНА-ЧАСТНОГО'
140 FOR I=N TO 0 STEP -1
150 PRINT!2.0!'A'I='!F1.9!A(I)
160 NEXT I:END

```

дробно-рациональной функции при заданном Z . Так, для приведенного примера МОДУЛЬ $= 0,04454466144$, ФАЗА $= -65,45048284^\circ$. Возможно также вычисление операторной функции $F(p)$, где $p=j\omega$ (вводится $\operatorname{Re} Z=0$ и $\operatorname{Im} Z=\omega$), а также вычисление функции $F(Z)=1/B(Z)$ (вводится $m=1$, $a_1=0$ и $a_0=1$). Для вычисления дробно-рациональной функции вводится $\operatorname{Re} Z=x$ и $\operatorname{Im} Z=0$.

Деление степенного многочлена $A(x)$ на двухчлен $B(x) = (x^2 + b_1x + b_0)$ выполняется для понижения степени исходного многочлена $A(x)$. Если $A(x)$ — многочлен четной степени, то из него можно выделить квад-

ратичный множитель $B(x)$, и использовать операцию деления для исключения двух корней, полученных из решения уравнения $B(x)=0$. В общем случае деление $A(x)$ на $B(x)$ дает многочлен $D(x)$ и остаток $R(x) = r_1x + r_0$.

Программа 3.34.

Пример. Для выражения

$$\frac{A(x)}{B(x)} = \frac{1x^6 + 7x^5 + 15x^4 + 15x^3 + 4x^2 - 3x + 125}{x^2 - 2x + 2}$$

получим $D(x) = 1x^4 + 9x^3 + 31x^2 + 59x + 60$ при остатке $R(x) = -1x + 5$.

```

05 PRINT 'ВЫЧИСЛЕНИЕ P(C) И ДЕЛЕНИЕ P(X) НА (X-C)'
10 INPUT 'ВВЕДИТЕ ЗНАЧЕНИЯ N,C 'N,C: DIM A(N)
20 FOR I=N TO 0 STEP -1
30 PRINT!2.0!'A'I'=': INPUT A(I): NEXT I
40 PRINT!2.0!'B'N-1'='!F1.9!A(N)
50 FOR I=N-1 TO 0 STEP -1
60 LET A(I)=A(I)+A(I+1)*C
70 IF I<>0 THEN PRINT!2.0!'B'I-1'='!F1.9!A(I)
80 NEXT I: PRINT!F1.9!'P(X)='A(0): END
    
```

Программа 3.36.

```

10 PRINT 'УМНОЖЕНИЕ МНОГОЧЛЕНА A(X) СТЕПЕНИ M'
20 PRINT ' НА МНОГОЧЛЕН B(X) СТЕПЕНИ N'
30 INPUT 'ВВЕДИТЕ M,N 'M,N: DIM A(M+M+1)
40 FOR I=M TO 0 STEP -1
50 PRINT!2.0!'A'I'=': INPUT A(I): NEXT I
60 FOR I=N+M TO M STEP -1
70 PRINT!2.0!'B'I-M'=': INPUT W
80 FOR J=M TO 1 STEP -1
90 LET A(J+M+1)=A(J)*W+A(J+M): NEXT J
100 LET A(M+1)=A(0)*W
110 PRINT!2.0!'C'I'='!F1.9!A(M+M+1): NEXT I
120 FOR I=M-1 TO 0 STEP -1
130 PRINT!2.0!'C'I'='!F1.9!A(I+M+1)
140 NEXT I: PRINT 'КОНЕЦ ВЫЧИСЛЕНИЙ': END
    
```

Вычисление значения полинома $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ в точке $x=c$ и деление $P(x)$ на $(x-c)$ выполняется по следующему алгоритму.

1. Вводим коэффициенты $a_n, a_{n-1}, \dots, a_1, a_0$ полинома $P(x)$.

2. Задаем $a_i = a_n$ ($i=n$).

3. Преобразуем a_i по рекуррентной формуле $a_i = a_i + a_{i-1}c$ с помощью цикла с управляющей переменной i , меняющейся от значения $n-1$ до 0 с шагом -1 .

4. На каждом цикле п. 3, кроме $i=0$, выводим значения $B_i = a_{i-1}$ полинома $B(x) = A(x)/(x-c) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0$, имеющего порядок $(n-1)$.

5. По окончании цикла по п. 3, 4 выводим $P(x) = a_0$.

Если c — корень $P(x)$, то деление $P(x)$ на $(x-c)$ может применяться для понижения степени многочлена и поиска его очередного корня.

Программа 3.35.

Пример. Для многочлена $P(x) = 3x^3 + 2x^2 - 5x + 7$ при $n=3$ и $c=3$ получим $B(x) = 3x^2 + 11x + 28$ и $P(c) = 91$.

Умножение многочлена $A(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0$ степени m на многочлен $B(x) = b_n x^n + b_{n-1} x^{n-1} + \dots + b_1 x + b_0$ дает многочлен $C(x) = c_{m+n} x^{m+n} + c_{m+n-1} x^{m+n-1} + \dots + c_1 x + c_0$ степени $(m+n)$. Эта операция выполняется по следующему алгоритму.

1. Вводим коэффициенты $a_m, a_{m-1}, \dots, a_1, a_0$.

2. Организуем внешний цикл вычислений c_i , задав начальное значение управляющей переменной цикла $i=(n+m)$ и конечное значение m с шагом -1 . В начале каждого цикла вводим b_{i-m} и переходим к п. 3.

3. Организуем внутренний цикл с управляющей переменной j , меняющейся от $j=m$ до 1 с шагом 1. В этом цикле преобразуем коэффициенты многочлена $A(x)$ по формуле $a_{j+m+1} = a_j b_{i-m} + a_{j+m}$.

4. При выходе из внутреннего цикла вычисляем $a_{m+1} = a_0 b_{i-m}$, выводим на печать $c_i = a_{2m+1}$ и переходим к п. 2 (т. е. завершаем внешний цикл).

5. С помощью цикла с управляющей переменной i , меняющейся от $i=m-1$ до 0 с шагом -1 , выводим на печать последние m значений $c_i = a_{i+m+1}$.

Программа 3.36.

Пример. Для $A(x) = 4x^4 + 3x^2 + 2x + 1x + 0,5$ и $B(x) = 2x^2 + 1x + 0,4$ получим многочлен $C(x) = 8x^6 + 10x^5 + 8,6x^4 + 5,2x^3 + 2,8x^2 + 0,9x + 0,2$.

§ 3.4. Вычисление ортогональных многочленов

Ортогональные многочлены Чебышева $T_n(x) = \omega(x)$ являются решениями дифференциального уравнения второго порядка

$$(1-x^2) \frac{d^2 \omega}{dx^2} - x \frac{d\omega}{dx} + n^2 \omega = 0. \quad (3.2)$$

Они могут вычисляться с помощью явного выражения

$$T_n(x) = \cos(n \arccos x) \quad (3.3)$$

или по рекуррентным формулам $T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x)$ при $T_0(x) = 1$ и $T_1(x) = x$. Дифференциальному уравнению (3.2) соответствуют и решения в виде функций

Программа 3.37.

```

10 PRINT 'ВЫЧИСЛЕНИЕ МНОГОЧЛЕНА ЧЕБЫШЕВА'
15 INPUT 'ВВЕДИТЕ N='N
20 INPUT 'МНОГОЧЛЕН 1-ГО ИЛИ 2-ГО РОДА' R
30 INPUT 'ВВЕДИТЕ X='X
40 IF R=2 THEN 60
50 PRINT 'TN(X)='COS(N*ACS(X)) : GOTO 30
60 PRINT 'UN(X)='SIN((N+1)*ACS(X))/SIN(ACS(X))
70 GOTO 30 : END

```

Программа 3.38.

```

10 PRINT 'ВЫЧИСЛЕНИЕ МНОГОЧЛЕНА ЛЕЖАНДРА'
20 INPUT 'ВВЕДИТЕ N>=2 N='N
25 INPUT 'ВВЕДИТЕ X='X:LETA=1:LETB=X
30 FOR I=1 TO N-1
40 LETP=((2*I+1)*X*B-I*A)/(I+1)
50 LETA=B:LETB=P:NEXT I
60 PRINT 'PN(X)='P:GOTO 25:END

```

Программа 3.39.

```

10 PRINT 'ВЫЧИСЛЕНИЕ МНОГОЧЛЕНА ЛАГЕРРА'
20 INPUT 'ВВЕДИТЕ N>=2 N='N
30 INPUT 'ВВЕДИТЕ X='X : LET A=1 : LET B=1-X
40 FOR I=1 TO N-1
50 LET L=((2*I+1-X)*B-I*A)/(I+1)
60 LET A=B : LET B=L : NEXT I
70 PRINT 'LN(X)='L : GOTO 30 : END

```

Бесселя второго рода $U_n(x)$:

$$U_0(x) = \arcsin x, \quad (3.4)$$

$$U_n(x) = \sin(n \arccos x) = \frac{\sqrt{1-x^2}}{n} \frac{d}{dx} T_n(x).$$

Они также могут вычисляться с помощью рекуррентных формул

$$U_{i+1} = 2xU_i(x) - U_{i-1}(x), \quad i=1, 2, \dots, n-1,$$

при $U_0(x) = 1$ и $U_1(x) = 2x$.

Поскольку в большинстве версий языка бейсик предусмотрено вычисление обратных тригонометрических функций, вычисление $T_n(x)$ и $U_n(x)$ целесообразно выполнять по явным формулам (3.3) и (3.4).

Программа 3.37.

Контрольные примеры. $T_3(0,1) = -0,26$; $U_3(0,1) = -0,392$.

Ортогональные многочлены Лежандра $P_n(x)$ и $U_n(x)$ являются решениями дифференциального уравнения

$$(1-x^2) \frac{d^2 \omega}{dx^2} - 2x \frac{d\omega}{dx} + n(n+1)\omega = 0$$

и могут вычисляться по рекуррентным формулам

$$P_{i+1}(x) = ((2i+1)xP_i(x) - iP_{i-1}(x))/(i+1),$$

$i=1, 2, \dots, n-1,$

при $P_0(x) = 1$ и $P_1(x) = x$.

Программа 3.38.

Пример. Для $n=4$ и $x=3$ $P_4(x) = 55,375$.

Ортогональные многочлены Лагерра $L_n(x)$ являются решениями дифференциального

уравнения

$$x \frac{d^2 \omega}{dx^2} + (1-x) \frac{d\omega}{dx} + n\omega = 0$$

и вычисляются с помощью рекуррентных формул

$$L_{i+1}(x) = ((2i+1-x)L_i(x) - iL_{i-1}(x))/(i+1),$$

$$i=1, 2, \dots, n-1,$$

при $L_0(x) = 1$ и $L_1(x) = 1-x$.

Программа 3.39.

Пример. $L_{10}(3) = -0,7000223214$;
 $L_{10}(10) = 27,98412698$.

Ортогональные многочлены Эрмита $H_n(x)$ соответствуют решениям дифференциального уравнения

$$\frac{d^2 \omega}{dx^2} - 2x \frac{d\omega}{dx} + 2n\omega = 0$$

и вычисляются по рекуррентным формулам

$$H_{i+1}(x) = 2xH_i(x) - 2iH_{i-1}(x),$$

$$i=1, 2, \dots, n-1,$$

при $H_0(x) = 1$, $H_1(x) = 2x$.

Программа 3.40.

```

10 PRINT 'ВЫЧИСЛЕНИЕ МНОГОЧЛЕНА ЭРМИТА'
20 INPUT 'ВВЕДИТЕ N>=2 N='N
30 INPUT 'ВВЕДИТЕ X='X:LETA=1:LETB=2*X
40 FOR I=1 TO N-1
50 LETH=2*X*B-2*I*A
60 LETA=B:LETB=H:NEXT I
70 PRINT 'HN(X)='H:GOTO 30:END

```

Пример. $H_{10}(3) = -3093984$, $H_{10}(10) = 8,09327821 \cdot 10^{12}$.

§ 3.5. Операции с матрицами

Матрица — двумерный массив чисел вида

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix} \quad (3.5)$$

Матрица характеризуется *размерностью* $m \times n$, т. е. произведением числа столбцов n на число строк m . *Квадратная матрица* имеет $m = n$ и размерность $n^2 = n \times n$. У простейших ПЭВМ предусмотрено формирование одной-двух матриц размерностью 10×10 с коэффициентами A_{ij} (например, у ПЭВМ класса Pocket Computers FX-702P можно формировать одну матрицу 10×20 или две матрицы 10×10 , причем нумерация строк и столбцов начинается с 0, а не с 1). У более сложных ПЭВМ коэффициенты матрицы могут задаваться любой переменной, например B_{ij} , Z_{kl} и т. д.

Ввод матрицы обычно задается по строкам с помощью двух циклов.

1. Задаем значения I от 1 (или 0) до M (или $M-1$) во внешнем цикле.
2. Задаем значения J от 1 (или 0) до N (или $N-1$) во внутреннем цикле.
3. С помощью оператора INPUT задаем ввод $A(I, J)$ каждого члена матрицы.

Перед операторами ввода матрицы необходимо задать размерность массива, например, с помощью операторов DIMA (M, N), SOMA (M, N) или других (зависят от конкретной версии бейсика, применяемой в ЭВМ).

Программа 3.41.

```
10 REM 'ВВОД ЭЛЕМЕНТОВ МАТРИЦЫ M*N
20 INPUT 'ВВЕДИТЕ M,N 'M,N: DIM A(M,N)
30 FOR I=1 TO M:FOR J=1 TO N
40 PRINT!2.0!'ВВЕДИТЕ ЭЛЕМЕНТ A'I','J
50 INPUT A(I,J):NEXT J:NEXT I
500 END
```

В ЭВМ с версией бейсика, имеющей матричные операторы, программа ввода существенно упрощается (см. пример в § 2.2).

Диагональная матрица является разновидностью квадратной матрицы вида

$$A = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}$$

У нее все элементы нулевые, кроме диагональных, т. е. $A(I, J) \neq 0$ при $I=J$.

Программа 3.42.

```
10 REM 'ФОРМИРОВАНИЕ ДИАГОНАЛЬНОЙ МАТРИЦЫ'
20 INPUT 'ВВЕДИТЕ N='N: DIM A(N,N)
30 FOR I=1 TO N:FOR J=1 TO N
40 IF I<>J THEN 60
50 PRINT!2.0!'ВВЕДИТЕ A'I','J
55 INPUT A(I,J):GOTO 70
60 LET A(I,J)=0
70 NEXT J:NEXT I
500 END
```

Единичная матрица — разновидность диагональной матрицы, у которой $A(I, J) = 1$ при $I=J$.

Программа 3.43.

```
10 REM 'ФОРМИРОВАНИЕ ЕДИНИЧНОЙ МАТРИЦЫ'
20 INPUT 'ВВЕДИТЕ N='N: DIM A(N,N)
30 FOR I=1 TO N:FOR J=1 TO N
40 IF I<>J THEN 60
50 LET A(I,J)=1:GOTO 70
60 LET A(I,J)=0
70 NEXT J:NEXT I
500 END
```

Нулевая матрица — очищенный массив, т. е. матрица, у которой все $A(I, J) = 0$. Для получения такой матрицы иногда могут использоваться специальные операторы полной очистки памяти данных (например, VAC у ПЭВМ FX-702P), исполняемые до ввода всех исходных данных. Если нулевую матрицу нужно формировать по ходу вычислений, можно использовать следующую программу:

Программа 3.44.

```
10 REM 'ФОРМИРОВАНИЕ НУЛЕВОЙ МАТРИЦЫ'
20 INPUT 'ВВЕДИТЕ M,N 'M,N: DIM A(M,N)
30 FOR I=1 TO M:FOR J=1 TO N
40 LET A(I,J)=0:NEXT J:NEXT I
500 END
```

Операции с константами сводятся к сложению или вычитанию элементов матрицы $A(I, J)$ с константой, например X , или умножению $A(I, J) * X$ или делению $A(I, J) / X$ на константу. Все они могут проводиться в ходе ввода матрицы (см. программу 3.45).

Программа 3.45.

```
10 PRINT 'УМНОЖЕНИЕ МАТРИЦЫ M*N
15 PRINT 'НА КОНСТАНТУ C'
20 INPUT 'ВВЕДИТЕ N='N
25 INPUT 'ВВЕДИТЕ C:C: DIM A(N,N)
30 FOR I=1 TO N: FOR J=1 TO N
40 PRINT!2.0!'ВВЕДИТЕ ЭЛЕМЕНТ A'I','J
50 INPUT A(I,J): LET A(I,J)=A(I,J)*C
60 NEXT J:NEXT I
500 END
```

Вывод матрицы на печать организуется, как и ввод, с помощью двух циклов. Однако вместо оператора ввода во внутреннем цикле используется оператор вывода PRINT. Целесообразно вывод организовать так, чтобы индцировались индексы I и J каждого элемента в формате без дробных частей (см. пример вывода для программы 3.45).

Программа 3.46.

```
70 FOR I=1 TO N:FOR J=1 TO N
80 PRINT!2.0!'A'I','J'='A(I,J)
90 NEXT J:NEXT I
500 END
```

Пример. Для $X=2$ и матрицы

$$A = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

выполнение совместно программ 3.45 и 3.46

даст такие результаты (элементы выводятся построчно):

$$\begin{array}{rcl} A & 1, & 1 = 2 \\ A & 1, & 2 = 4 \\ \dots & \dots & \dots \\ A & 3, & 3 = 18 \end{array}$$

Для матриц небольшой размерности вывод можно задать и в естественном для матрицы виде.

Программа 3.47.

```
70 PRINT A(1,1),A(1,2),A(1,3)
80 PRINT A(2,1),A(2,2),A(2,3)
90 PRINT A(3,1),A(3,2),A(3,3)
500 END
```

Пример. Для приведенного выше примера совместное выполнение программ 3.45 и 3.47 ведет к выдаче результатов в форме

$$\begin{array}{ccc} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{array}$$

Транспонированной матрицей A^T называется квадратная матрица, у которой столбцы соответствуют строкам квадратной матрицы A , например:

$$A = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

$$A^T = \begin{vmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{vmatrix}$$

Нетрудно заметить, что диагональные элементы у матриц A и A^T одни и те же и операция транспонирования матрицы A сводится к перестановке остальных элементов симметрично относительно диагонали. Это реализуется с помощью соотношения $a_{ij} = a_{ji}$ при $i \neq j$.

Программа 3.48.

```
10 PRINT 'ТРАНСПОНИРОВАНИЕ МАТРИЦЫ'
20 INPUT 'ВВЕДИТЕ N=N: DIM A(N,N), B(N,N)'
30 FOR I=1 TO N: FOR J=1 TO N
40 PRINT!Z.0!'ВВЕДИТЕ A'I', 'J': INPUT A(I, J)
50 NEXT J: NEXT I
60 FOR I=1 TO N: FOR J=I+1 TO N
70 LET C=A(I, J): LET A(I, J)=A(J, I): LET A(J, I)=C
80 NEXT J: NEXT I
90 REM 'ТРАНСПОНИРОВАННАЯ МАТРИЦА ЗАНЕСЕНА НА МЕСТО ИСХОДНОЙ'
500 END
```

Определитель квадратной матрицы A — число D , равное сумме $n!$ членов $(-1)^{i_1 i_2 \dots i_n} a_{1i_1} a_{2i_2} a_{3i_3} \dots a_{ni_n}$, каждый из которых соответствует одному из $n!$ различных упорядоченных множеств j_1, j_2, \dots, j_n , полученных r -парными перестановками элементов из множества $1, 2, \dots, n$.

Определители матриц малой размерности (2×2 , 3×3 , 4×4) целесообразно вычислять по известным формулам:

$$D = a_{11}a_{22} - a_{21}a_{12}$$

для матрицы 2×2 :

$$D = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{12}a_{23}a_{31} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{41} - a_{21}a_{12}a_{33}$$

для матрицы 3×3 :

$$D = (a_{11}a_{22} - a_{12}a_{21})(a_{33}a_{44} - a_{34}a_{43}) + (a_{13}a_{21} - a_{11}a_{23})(a_{32}a_{44} - a_{34}a_{42}) + (a_{11}a_{24} - a_{14}a_{21})(a_{32}a_{43} - a_{33}a_{42}) + (a_{12}a_{23} - a_{13}a_{22})(a_{31}a_{44} - a_{34}a_{41}) + (a_{14}a_{22} - a_{12}a_{24})(a_{31}a_{43} - a_{33}a_{41}) + (a_{13}a_{24} - a_{14}a_{23})(a_{31}a_{42} - a_{32}a_{41})$$

для матрицы 4×4 .

Программа 3.49.

Пример.

$$D = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = -2$$

Программа 3.50.

Пример.

$$D = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & -9 \end{vmatrix} = 54.$$

Программа 3.51.

Пример.

$$D = \begin{vmatrix} 2 & 0 & 1 & 3 \\ -1 & 2 & -1 & 1 \\ 0 & 1 & 0 & -2 \\ 1 & -3 & 2 & 1 \end{vmatrix} = 18.$$

Определители матриц с любой размерностью $n \times n$ вычисляются по методу Гаусса. Он сводится к преобразованию матриц к треугольному виду с помощью следующих формул преобразования элементов матрицы A (3.5):

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$$

где $k=1, 2, \dots, (n-1)$ и $a_{kk}^{(k-1)} \neq 0$. Преобразование массива $A(n, n)$ производится в направлении расположения столбцов слева

направо. Определитель вычисляется как произведение всех диагональных элементов преобразованной матрицы.

Программа 3.52.

Проверка программы производится по примерам, приведенным выше для программ 3.49 и 3.50.

Необходимое условие для реализации простого метода Гаусса заключается в неравенстве элемента $a_{kk}^{(k-1)}$ нулю на всех этапах преобразования. В случае $a_{kk}^{(k-1)} = 0$ произойдет останов ЭВМ, так как деление

Программа 3.49.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЯ 2*2'
20 INPUT 'ВВЕДИТЕ A11,A12,A21,A22 'A,B,C,D
30 PRINT 'ОПРЕДЕЛИТЕЛЬ D='A*D-C*B:END

```

Программа 3.50.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЯ 3*3'
20 INPUT 'ВВЕДИТЕ A11,A12,A13 'A,B,C
30 INPUT 'ВВЕДИТЕ A21,A22,A23 'D,E,F
40 INPUT 'ВВЕДИТЕ A31,A32,A33 'G,H,I
50 LETS=A*(E*I-H*F)+C*(D*H-G*E)+B*(F*G-D*I)
60 PRINT 'ОПРЕДЕЛИТЕЛЬ D='S:END

```

Программа 3.51

```

10 PRINT 'ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЯ 4*4'
15 DIM A(4,4):LETS=0
20 FOR I=1 TO 4:FOR J=1 TO 4
30 PRINT 'ВВЕДИТЕ ЭЛЕМЕНТ A='!2.0!I',J
40 INPUT A(I,J):NEXT J:NEXT I
50 LETO=1:LETP=2:LETO=3:LETR=4:GOSUB 120
60 LETO=3:LETP=1:LETO=2:LETR=4:GOSUB 120
70 LETO=1:LETP=4:LETO=2:LETR=3:GOSUB 120
80 LETO=2:LETP=3:LETO=1:LETR=4:GOSUB 120
90 LETO=4:LETP=2:LETO=1:LETR=3:GOSUB 120
100 LETO=3:LETP=4:LETO=1:LETR=2:GOSUB 120
110 PRINT '!F1.9!' 'ОПРЕДЕЛИТЕЛЬ D='S:STOP
120 LETB=A(1,0)*A(2,P)-A(1,P)*A(2,0)
130 LETS=S+B*(A(3,0)*A(4,R)-A(3,R)*A(4,0)).
140 RETURN:END

```

Программа 3.52.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЯ'
20 INPUT 'ВВЕДИТЕ ПОРЯДОК ОПРЕДЕЛИТЕЛЯ N='N
30 DIM A(N,N):LETP=1
40 FOR I=1 TO N:FOR J=1 TO N
50 PRINT '!2.0!' 'ВВЕДИТЕ ЭЛЕМЕНТ A' I',J
60 INPUT A(I,J):NEXT J:NEXT I
70 FOR I=1 TO N-1:FOR J=I+1 TO N
80 LETA(J,I)=A(J,I)/A(I,I):FOR K=I+1 TO N
90 LETA(J,K)=A(J,K)-A(J,I)*A(I,K):NEXT K
100 NEXT J:LETP=P*A(I,I):NEXT I
110 LETP=P*A(N,N):PRINT '!F1.9!' 'ОПРЕДЕЛИТЕЛЬ D='P:END

```

Программа 3.53.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЯ ПО МЕТОДУ ГАУССА'
15 PRINT 'С ВЫБОРОМ ГЛАВНОГО ЭЛЕМЕНТА ПО ВСЕЙ МАТРИЦЕ'
20 INPUT 'ЗАДАЙТЕ ЗНАЧЕНИЕ N='N:DIM A(N,N)
25 FOR I=0 TO N-1:FOR J=0 TO N-1
30 PRINT '!2.0!' 'ВВЕДИТЕ ЭЛЕМЕНТ A' I+1',J+1'='
40 INPUT A(I,J):NEXT J:NEXT I:LETP=0
50 LETZ=1:LETD=1:FOR K=0 TO N-2
60 LETE=0:FOR I=K TO N-1
70 FOR J=K TO N-1:IF ABS(E)>=ABS(A(I,J)) THEN 90
80 LETE=A(I,J):LETB=I:LETC=J:NEXT J:NEXT I
90 IF K=B THEN 120
100 FOR J=K TO N-1:LETS=A(K,J)
110 LETA(K,J)=A(B,J):LETA(B,J)=S:NEXT J:LETZ=-Z
120 IF K=C THEN 150
125 FOR I=K TO N-1
130 LETS=A(I,K):LETA(I,K)=A(I,C)
140 LETA(I,C)=S:NEXT I:LETZ=-Z
150 FOR I=K+1 TO N-1:LETB=A(I,K)/A(K,K)
160 FOR J=K TO N-1:LETA(I,J)=A(I,J)-B*A(K,J)
170 NEXT J:NEXT I:NEXT K
180 FOR I=0 TO N-1:LETD=D*A(I,I)
190 NEXT I:LETD=D*Z:PRINT 'ОПРЕДЕЛИТЕЛЬ D='!F1.9!D:END

```


на 0 считается некорректной операцией. Большие погрешности возникают, если $a^{(k-1)} \neq 0$, но является малым числом.

Для повышения точности вычисления определителя и исключения останова ЭВМ в ходе вычислений применяются варианты метода Гаусса с выбором главного элемента по столбцу, строке или по всей матрице. В последнем случае перед преобразованиями в матрице A (3.5) выбирают наибольший по модулю (главный) элемент, допустим a_{ij} . Далее переставляют первую строку матрицы с i -й строкой (это и есть выбор главного элемента по строке), а первый столбец переставляют с j -м (выбор главного элемента по столбцу). При каждой такой перестановке знак определителя меняется, поэтому результат нужно умножить на $(-1)^P$, где P — число перестановок.

Программа 3.53.

Пример.

$$D = \begin{vmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 6 \\ 1 & 2 & 4 & 8 \\ 9 & 8 & 7 & 6 \end{vmatrix} = -20.$$

Отметим, что вычисление этого определителя по программе 3.52 ведет к останову ЭВМ с индикацией ошибки (некорректная математическая операция).

Обратной матрицей называется матрица A^{-1} , произведение которой на заданную матрицу A дает единичную матрицу.

Обращение матрицы A заключается в нахождении обратной матрицы A^{-1} . Для обращения матрицы A используют следующие соотношения:

$$A \cdot A^{-1} = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = \begin{vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{vmatrix} = \begin{vmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{vmatrix}$$

Перемножая матрицы A и A^{-1} , получим n систем уравнений с n^2 неизвестными:

$$\begin{aligned} a_{11}x_{11} + a_{12}x_{21} + \dots + a_{1n}x_{n1} &= 1, \\ a_{21}x_{11} + a_{22}x_{21} + \dots + a_{2n}x_{n1} &= 0, \\ a_{n1}x_{11} + a_{n2}x_{21} + \dots + a_{nn}x_{n1} &= 0. \end{aligned} \quad (1)$$

$$\begin{aligned} a_{11}x_{12} + a_{12}x_{22} + \dots + a_{1n}x_{n2} &= 0, \\ a_{21}x_{12} + a_{22}x_{22} + \dots + a_{2n}x_{n2} &= 1, \\ a_{n1}x_{12} + a_{n2}x_{22} + \dots + a_{nn}x_{n2} &= 0. \end{aligned} \quad (2)$$

$$\begin{aligned} a_{11}x_{1n} + a_{12}x_{2n} + \dots + a_{1n}x_{nn} &= 0, \\ a_{21}x_{1n} + a_{22}x_{2n} + \dots + a_{2n}x_{nn} &= 0, \\ a_{n1}x_{1n} + a_{n2}x_{2n} + \dots + a_{nn}x_{nn} &= 1. \end{aligned} \quad (n)$$

Решение этих уравнений определяет значения всех элементов матрицы $A^{-1}x_{ij}$. Оно выполняется по методам, описанным в § 4.1. Для простых матриц обращение выполняется по программам, реализующим готовые «рецепты» процедуры обращения.

Программа 3.54.

```
10 PRINT 'ОБРАЩЕНИЕ МАТРИЦЫ 2*2'
20 INPUT 'ВВЕДИТЕ A11, A12' A, B
25 INPUT 'ВВЕДИТЕ A21, A22' C, D
30 LET E=C*B: LET F=A: LET G=D*F-E: LET A=D/E
40 LET D=F/E: LET C=-C/E: LET B=-B/E
50 PRINT 'ОБРАЩЕННАЯ МАТРИЦА'
60 PRINT A, B: PRINT C, D: END
```

Пример.

$$A = \begin{vmatrix} 7 & 8 \\ 4 & 5 \end{vmatrix}, \quad A^{-1} = \begin{vmatrix} 1,66666667 & -2,66666667 \\ -1,33333333 & 2,33333333 \end{vmatrix}$$

Программа 3.55.

Пример.

$$A = \begin{vmatrix} 4 & 8 & 0 \\ 8 & 8 & 8 \\ 2 & 0 & 1 \end{vmatrix}, \quad A^{-1} = \begin{vmatrix} 0,0833333 & -0,0833333 & 0,666666 \\ 0,0833333 & 0,0416666 & -0,333333 \\ -0,166666 & 0,166666 & -0,333333 \end{vmatrix}$$

(узаными 5 значащими цифр результата).

Программа 3.56.

Проверить эту программу можно по примеру, приведенному к программе 3.55.

Сложение и вычитание для двух матриц A и B с одинаковой размерностью $m \times n$ производится по формуле

$$c_{ij} = a_{ij} \pm b_{ij},$$

где $i=1, 2, \dots, m$ и $j=1, 2, \dots, n$. Здесь c_{ij} — элементы результирующей матрицы C .

Программа 3.57.

```
05 PRINT 'СЛОЖЕНИЕ МАТРИЦ'
10 INPUT 'ВВЕДИТЕ M, N' M, N
15 DIM A(M, N), B(M, N), C(M, N)
20 FOR I=1 TO M: FOR J=1 TO N
30 PRINT!2, 0!'ВВЕДИТЕ A'I', 'J'
40 INPUT A(I, J): NEXT J: NEXT I
50 FOR I=1 TO M: FOR J=1 TO N
60 PRINT 'ВВЕДИТЕ B'I', 'J'
65 INPUT B(I, J)
70 LET C(I, J)=A(I, J)+B(I, J)
80 NEXT J: NEXT I: END
```

Пример. Для $m=2, n=3$

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{vmatrix} + \begin{vmatrix} 7 & 8 & 9 \\ 8 & 7 & 6 \end{vmatrix} = \begin{vmatrix} 8 & 10 & 11 \\ 12 & 12 & 12 \end{vmatrix}$$

Умножение матриц A с размерностью $m \times n$ и B с размерностью $n \times l$ выполняется по формуле

$$c_{kj} = \sum_{i=1}^n (a_{ki} b_{ij}),$$

где $j=1, 2, \dots, l$ и $k=1, 2, \dots, m$. Получаемая матрица C имеет размерность $m \times l$.

Программа 3.55

```

10 PRINT 'ОБРАЩЕНИЕ МАТРИЦЫ 3*3': DIM A(3,3)
15 INPUT 'ВВЕДИТЕ A11,A12,A13 'A(1,1),A(1,2),A(1,3)
20 INPUT 'ВВЕДИТЕ A21,A22,A23 'A(2,1),A(2,2),A(2,3)
25 INPUT 'ВВЕДИТЕ A31,A32,A33 'A(3,1),A(3,2),A(3,3)
30 FOR K=1 TO 3
35 LET A=A(2,1)/A(1,1): LET B=-A
40 LET C=A(2,2)-A*A(1,2): LET D=A(2,3)-A*A(1,3)
50 LET A=A(3,1)/A(1,1): LET A(2,3)=-A
60 LET A(2,1)=A(3,2)-A*A(1,2)
70 LET A(2,2)=A(3,3)-A*A(1,3)
80 LET A(3,1)=A(1,2)/A(1,1)
90 LET A(3,2)=A(1,3)/A(1,1)
100 LET A(3,3)=1/A(1,1): LET A(1,1)=C
110 LET A(1,2)=D: LET A(1,3)=B: NEXT K
120 PRINT 'ОБРАЩЕННАЯ МАТРИЦА'
130 PRINT A(1,1),A(1,2),A(1,3)
140 PRINT A(2,1),A(2,2),A(2,3)
150 PRINT A(3,1),A(3,2),A(3,3): END

```

Программа 3.56.

```

05 PRINT 'ОБРАЩЕНИЕ МАТРИЦ МЕТОДОМ ГАУССА'
10 PRINT 'С ВЫБОРОМ ГЛАВНОГО ЭЛЕМЕНТА'
15 INPUT 'ЗАДАЙТЕ N='N': DIM A(N,N),P(N,N)
20 DIM B(N),C(N,N),G(N),X(N)
25 FOR I=1 TO N: FOR J=1 TO N
30 PRINT !2.0! 'ВВЕДИТЕ A' I ', ' J '= ' : INPUT A(I,J)
35 LET P(I,J)=A(I,J): NEXT J: NEXT I
40 FOR J2=1 TO N: FOR I=1 TO N
50 LET B(I)=0: NEXT I
60 LET B(J2)=1
70 FOR J3=1 TO N: FOR J4=1 TO N
80 LET A(J3,J4)=P(J3,J4): NEXT J4: NEXT J3
85 GOSUB 100: FOR I=1 TO N
90 PRINT !2.0! 'X' I ', ' J2 '= ' !F1.9!X(I)
95 NEXT I: NEXT J2: STOP
100 LET N1=N-1: FOR K=1 TO N1
110 IF ABS(A(K,K))>0 GOTO 200
120 LET K1=K+1: FOR M=K1 TO N
130 IF ABS(A(M,K))>0 GOTO 150
140 GOTO 165
150 FOR L=1 TO N: LET U=A(K,L): LET A(K,L)=A(M,L)
160 LET A(M,L)=U: NEXT L: NEXT M
170 LET U=B(K): LET B(K)=B(M): LET B(M)=U
200 LET G(K)=B(K)/A(K,K): LET K1=K+1
210 FOR I=K1 TO N: LET B(I)=B(I)-A(I,K)*G(K)
220 FOR J1=K TO N: LET J=N-J1+K: LET C(K,J)=A(K,J)/A(K,K)
225 LET A(I,J)=A(I,J)-A(I,K)*C(K,J)
230 NEXT J1: NEXT I: NEXT K
240 LET M=N: LET X(M)=B(M)/A(M,M)
250 LET M=M-1: LET S=0: FOR L=M TO N1
260 LET S=S+C(M,L+1)*X(L+1): NEXT L
270 LET X(M)=G(M)-S: IF M>1 GOTO 250
280 RETURN: END

```

Программа 3.58.

```

10 PRINT 'УМНОЖЕНИЕ МАТРИЦ M*N И N*L'
20 INPUT 'ВВЕДИТЕ M,N,L' M,N,L
30 DIM A(M,N),B(N,L),C(N,L)
40 FOR I=1 TO M: FOR J=1 TO N
50 PRINT !2.0! 'ВВЕДИТЕ A' I ', ' J ': INPUT A(I,J)
60 NEXT J: NEXT I
70 FOR I=1 TO N: FOR J=1 TO L
80 PRINT 'ВВЕДИТЕ B' I ', ' J ': INPUT B(I,J)
90 NEXT J: NEXT I
100 FOR K=1 TO M: FOR J=1 TO L: LET S=0
110 FOR I=1 TO N: LET S=S+A(K,I)*B(I,J): NEXT I
120 LET C(K,J)=S: NEXT J: NEXT K
130 PRINT 'РЕЗУЛЬТАТ'
140 FOR K=1 TO M: FOR J=1 TO L
150 PRINT !2.0! 'C' K ', ' J '= ' !F1.9!C(K,J)
160 NEXT J: NEXT K: END

```

Программа 3.58.

Пример. Для $m=4, n=5, l=3$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 1 & 2 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 15 & 30 & 45 \\ 31 & 62 & 93 \\ 20 & 40 & 81 \\ 27 & 54 & 81 \end{pmatrix}$$

Умножение матрицы на вектор $R = \{r_1, r_2, \dots, r_m\}$ выполняется по формуле

$$U_i = \sum_{j=1}^n a_{ij} r_j,$$

где $i=1, 2, \dots, m$. В результате получаем вектор $U = \{u_1, u_2, \dots, u_m\}$.

Скалярное произведение векторов $A = \{a_1, a_2, \dots, a_n\}$ и $B = \{b_1, b_2, \dots, b_n\}$ есть число

$$C = \sum_{i=1}^n a_i b_i,$$

где $i=1, 2, \dots, n$.

Ввиду простоты составления программ для двух последних операций они не приводятся.

§ 3.6. Вычисление факториалов и комбинаторика

Факториал при целом $n \geq 1$

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1 \quad (3.6)$$

с особым случаем $0! = 1$ у ряда ПЭВМ может вычисляться микропрограммно (например, с помощью оператора ! у ПЭВМ FX-702P). Если такой возможности нет, вычисление факториала проводится по формуле (3.6).

Программа 3.60.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ N!!'
10 INPUT 'ВВЕДИТЕ ЧИСЛО N=':N:LETA=N/2:LETB=1
15 FOR I=1 TO 10
20 IF A-INT(A)=0 THEN 60
30 FOR I=1 TO N STEP 2
40 LETB=B*I:NEXT I
50 GOTO 80
60 FOR J=2 TO N STEP 2
70 LETB=B*J:NEXT J
80 PRINT 'ЗНАЧЕНИЕ N!!=':B
90 GOTO 10:END
```

Программа 3.61.

```
10 PRINT 'РАСЧЕТ ЧИСЛА ПЕРЕСТАНОВОК, РАЗМЕШЕНИЙ'
20 PRINT 'И СОЧЕТАНИЙ ИЗ N ЭЛЕМЕНТОВ ПО M'
30 INPUT 'ВВЕДИТЕ ЦЕЛЫЕ ЧИСЛА N, M':N,M
40 LETX=N:GOSUB 100:LETR=R
50 LETX=N-M:GOSUB 100:LETA=P/R
60 LETX=M:GOSUB 100:LETC=A/R
70 PRINT 'ЧИСЛО ПЕРЕСТАНОВОК PN=':P
80 PRINT 'ЧИСЛО РАЗМЕШЕНИЙ A M,N=':A
90 PRINT 'ЧИСЛО СОЧЕТАНИЙ C M,N=':C:STOP
100 LETR=1:FOR I=X TO 1 STEP -1
110 LETR=R*I:NEXT I:RETURN:END
```

Программа 3.59.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФАКТОРИАЛА'
10 INPUT 'ВВЕДИТЕ ЧИСЛО ЧИСЕЛ N=':N
20 LET I=0: LET P=1
30 LET I=I+1: LET P=P*I
40 IF I<N THEN 30
50 PRINT 'ФАКТОРИАЛ N!=':P
70 GOTO 10:END
```

Пример $5! = 120, 10! = 3628800$.

При $n \geq 70$ значения $n!$ выходят за пределы разрядности большинства ПЭВМ ($n! > 10^{90}$).

Для приближенного вычисления факториала больших чисел n применяется формула Стирлинга [10, 36].

Функция $n!!$ вычисляется по формуле

$$n!! = \begin{cases} n(n-2) \cdot \dots \cdot 1 & \text{при нечетном } n, \\ n(n-2) \cdot \dots \cdot 2 & \text{при четном } n, \end{cases}$$

Программа 3.60.

Пример $7!! = 105, 8!! = 384$.

Число перестановок $P_n = n!$, а число размещений из n элементов по m

$$A_n^m = n! / (n-m) \dots$$

Число сочетаний из n элементов по m

$$C_n^m = n! / ((n-m)!m!).$$

причем $n \geq 1$ и $1 \leq m \leq n$.

Программа 3.61.

Пример. Для $m=5$ и $n=10$ получим

$$P_{10} = 3628800, A_{10}^5 = 30240 \text{ и } C_{10}^5 = 252.$$

Подобным способом можно вычислять и другие функции, содержащие члены с факториалами.

§ 3.7. Преобразование координат и векторный анализ

Преобразование координат из декартовых в полярные на плоскости описывалось в § 2.3. Преобразование декартовых координат X, Y, Z в сферические в пространстве (рис. 3.1) проводится по формулам

$$R = \sqrt{X^2 + Y^2 + Z^2}, \quad Q = \arctg(Y/X), \\ \varphi = \arccos(Z/R),$$

где R — полярный радиус, Q — долгота и φ — широта. Обратные преобразования проводятся по формулам

$$X = R \sin \varphi \cos Q, \quad Y = R \sin \varphi \sin Q, \quad Z = R \cos \varphi.$$

Таким образом, если положить $Z=0, \varphi=90^\circ$, то из этих формул вытекают формулы для преобразования декартовых координат в полярные и наоборот на плоскости.

Программа 3.62.

```
10 PRINT 'ПРЕОБРАЗОВАНИЕ КООРДИНАТ В ПРОСТРАНСТВЕ'
20 PRINT 'СФЕРИЧЕСКИХ В ДЕКАРТОВЫ -КОД 0'
30 PRINT 'ДЕКАРТОВЫХ В СФЕРИЧЕСКИЕ -КОД 1'
40 INPUT 'ЗАДАЙТЕ КОД 'A': IF A=1 THEN 110
50 INPUT 'ВВЕДИТЕ ПОЛЯРНЫЙ РАДИУС R='R
60 INPUT 'ВВЕДИТЕ ДОЛГОТУ-УГОЛ(ГРАД) Q='Q: LET Q1=RAD(Q)
70 INPUT 'ВВЕДИТЕ ШИРОТУ-УГОЛ(ГРАД) F='F: LET F1=RAD(F)
80 LET X=R*SIN(F1)*COS(Q1): LET Y=R*SIN(F1)*SIN(Q1)
90 LET Z=R*COS(F1): PRINT 'ДЕКАРТОВЫ КООРДИНАТЫ'
100 PRINT 'X='X, 'Y='Y, 'Z='Z: STOP
110 INPUT 'ВВЕДИТЕ КООРДИНАТЫ X,Y,Z 'X,Y,Z
120 LET R=SQR(X*X+Y*Y+Z*Z): LET F=ACS(Z/R): LET Q=ATN(Y/X)
130 PRINT 'ПОЛЯРНЫЙ РАДИУС R='R, 'ДОЛГОТА Q='DEG(Q)'ГРАД'
140 PRINT 'ШИРОТА F='DEG(F)'ГРАД': END
```

Пример. Для $R=10, Q=40^\circ$ и $\varphi=F=60^\circ$ получим (код 0) $X=6,634139482, Y=5,566703992$ и $Z=5$, а для $X=Y=Z=10$ получим (код 1) $R=17,32050808, Q=45^\circ$ и $\varphi=F=54,73561032^\circ$.

Преобразование декартовых координат на плоскости при сдвиге и повороте координатных осей (рис. 3.2) осуществляется по формулам (далее $\theta=Q$)

$$X' = (X - X_0) \cos Q + (Y - Y_0) \sin Q, \\ Y' = (Y - Y_0) \cos Q - (X - X_0) \sin Q,$$

где X' и Y' — новые координаты точки (X, Y) , X_0 и Y_0 — координаты центра новой системы координат.

Программа 3.63.

```
10 PRINT 'ПРЕОБРАЗОВАНИЕ КООРДИНАТ НА ПЛОСКОСТИ'
20 INPUT 'ЗАДАЙТЕ УГОЛ ПОВОРОТА ПЛОСКОСТЕЙ (ГРАД) Q='Q
30 LET Q=RAD(Q): INPUT 'ЗАДАЙТЕ КООРДИНАТЫ ЦЕНТРА X0,Y0 'B,C
40 INPUT 'ВВЕДИТЕ СТАРЫЕ КООРДИНАТЫ X,Y 'X,Y
45 LET X1=(X-B)*COS(Q)+(Y-C)*SIN(Q)
50 LET Y1=(Y-C)*COS(Q)-(X-B)*SIN(Q)
60 PRINT 'НОВЫЕ КООРДИНАТЫ'
70 PRINT 'X1='X1, 'Y1='Y1: GOTO 40: END
```

Пример. Для $Q=20^\circ, X_0=3, Y_0=2, X=5$ и $Y=5$ получим

$$X' = 2,905445672 \quad \text{и} \quad Y' = 2,135037676.$$

Вектор (отрезок) с координатами начала (X_1, Y_1) и конца (X_2, Y_2) , лежащий на

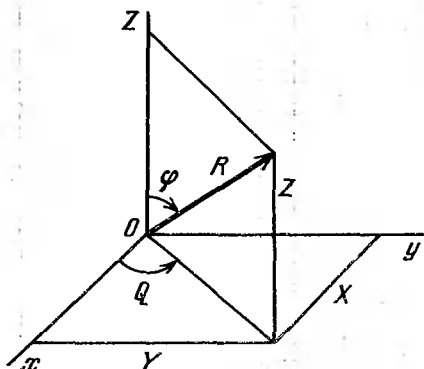


Рис. 3.1. Декартова и сферическая системы координат

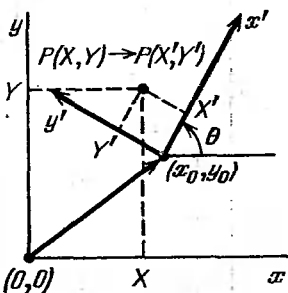


Рис. 3.2. Преобразование декартовых координат на плоскости при сдвиге и повороте осей

плоскости, имеет длину

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}.$$

Координаты точки (X, Y) , делящей этот отрезок в отношении M/N :

$$X = \frac{NX_1 + MX_2}{M+N}, \quad Y = \frac{NY_1 + MY_2}{M+N}.$$

Программа 3.64.

```

10 PRINT 'ДЛИНА ОТРЕЗКА И ЕГО ДЕЛЕНИЕ В ОТНОШЕНИИ M:N'
20 INPUT 'ЗАДАЙТЕ КООРДИНАТЫ ТОЧКИ 1 X1,Y1 'X1,Y1
30 INPUT 'ЗАДАЙТЕ КООРДИНАТЫ ТОЧКИ 2 X2,Y2 'X2,Y2
40 LET D=SQR((X2-X1)^2+(Y2-Y1)^2)
50 PRINT 'ДЛИНА ОТРЕЗКА D='D
60 INPUT 'ЗАДАЙТЕ M,N 'M,N
70 LET X=(N*X1+M*X2)/(M+N)
80 LET Y=(N*Y1+M*Y2)/(M+N)
90 PRINT 'КООРДИНАТЫ ТОЧКИ ДЕЛЕНИЯ ОТРЕЗКА'
100 PRINT 'X='X,'Y='Y:END
    
```

Пример. Для $X_1=2,3$, $Y_1=4$, $X_2=8,5$ и $Y_2=0,7$ получим $D=7,023531875$. Задав $M=3$ и $N=2$, получим $X=6,02$ и $Y=2,02$.

Полярными параметрами прямой $Ax + By + C = 0$, лежащей на плоскости (рис. 3.3), являются полярное расстояние от нее до точки $(0,0)$

$$P = |C| / \sqrt{A^2 + B^2}$$

и полярный угол Q , вычисляемый из соотношений

$$\cos Q = \mp A / \sqrt{A^2 + B^2},$$

$$\sin Q = \mp B / \sqrt{A^2 + B^2},$$

где знак $-$ берется, если $C > 0$, а знак $+$, если $C < 0$.

Программа 3.65.

```

10 PRINT 'ПОЛЯРНЫЕ ПАРАМЕТРЫ ПРЯМОЙ A*X+B*Y+C=0'
20 INPUT 'ВВЕДИТЕ A,B,C 'A,B,C
30 LET D=1: IF C>0 THEN LET D=-1
40 LET F=SQR(A*A+B*B): LET P=ABS(C)/F
50 PRINT 'КРАТЧАЙШЕЕ РАССТОЯНИЕ ДО ТОЧКИ (0,0) P='P
60 PRINT 'COS(Q)='D*A/F,'SIN(Q)='D*B/F
70 PRINT 'УГОЛ Q='DEG(ACS(D*A/F)):END
    
```

Пример. Для $A=3$, $B=-4$ и $C=10$ получим $P=2$, $\cos Q = -0,6$, $\sin Q = 0,8$ и $Q = 126,8698976^\circ$.

Вектор в пространстве с началом в точке $(0,0,0)$ декартовой системы координат и концом в точке (X, Y, Z) имеет длину

$$D = \sqrt{X^2 + Y^2 + Z^2}$$

и образует с осями координат углы

$$\alpha = \arccos(X/D), \quad \beta = \arccos(Y/D)$$

и

$$\gamma = \arccos(Z/D).$$

Программа 3.66.

```

10 PRINT 'ДЛИНА ВЕКТОРА И УГОЛ МЕЖДУ НИМ И КООРДИНАТНЫМИ ОСЯМИ'
20 INPUT 'ЗАДАЙТЕ КООРДИНАТЫ КОНЦА ВЕКТОРА X,Y,Z 'X,Y,Z
30 LET D=SQR(X*X+Y*Y+Z*Z): LET A=X/D: LET B=Y/D: LET C=Z/D
40 PRINT 'COS АЛЬФА='A,'COS БЕТА='B,'COS ГАММА='C
50 PRINT 'АЛЬФА='DEG(ACS(A)), 'БЕТА='DEG(ACS(B)),
60 PRINT 'ГАММА='DEG(ACS(C)):GOTO 20:END
    
```

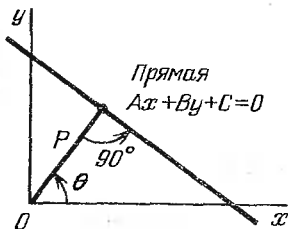


Рис. 3.3. Полярные параметры прямой на плоскости

Пример. Для $X=2$, $Y=-2$ и $Z=-1$ получаем $\cos \alpha = 0,666666667$, $\cos \beta = -0,666666667$, $\cos \gamma = -0,333333333$, $\alpha = 48,1896851^\circ$, $\beta = 131,8103149^\circ$ и $\gamma = 109,4712206^\circ$.

Два вектора с началом в точке $(0,0,0)$ и концами в точках (X_1, Y_1, Z_1) и (X_2, Y_2, Z_2) образуют между собой угол (рис. 3.4)

$$Q = \arccos \frac{X_1 X_2 + Y_1 Y_2 + Z_1 Z_2}{\sqrt{X_1^2 + Y_1^2 + Z_1^2} \sqrt{X_2^2 + Y_2^2 + Z_2^2}}$$

Длина отрезка, заданного концами этих векторов или просто точками с их-координатами,

$$D = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2}$$

Координаты точки (X, Y, Z) , делящей этот отрезок в отношении M/N , выражаются формулами

$$X = \frac{NX_1 + MX_2}{M + N}, \quad Y = \frac{NY_1 + MY_2}{M + N},$$

$$Z = \frac{NZ_1 + MZ_2}{M + N}$$

Программа 3.67.

```

10 PRINT 'УГОЛ МЕЖДУ ДВУМЯ ВЕКТОРАМИ, ДЛИНА ОТРЕЗКА МЕЖДУ'
20 PRINT 'ИХ КОНЦАМИ И КООРДИНАТЫ ТОЧКИ, ДЕЛЯЩЕЙ ОТРЕЗОК В'
30 PRINT 'ОТНОШЕНИИ M:N'
40 PRINT 'ВВЕДИТЕ КООРДИНАТЫ КОНЦА ПЕРВОГО ВЕКТОРА'
50 INPUT 'X1, Y1, Z1 'A, B, C
60 PRINT 'ВВЕДИТЕ КООРДИНАТЫ КОНЦА ВТОРОГО ВЕКТОРА'
70 INPUT 'X2, Y2, Z2 'D, E, F
80 LET S=SQR((A-D)^2+(B-E)^2+(C-F)^2)
90 LET Q=SQR(A*A+B*B+C*C)*SQR(D*D+E*E+F*F)
100 LET Q=(A*D+B*E+C*F)/Q
110 PRINT 'УГОЛ МЕЖДУ ВЕКТОРАМИ(ГРАД) Q='DEG(ACS(Q))
120 PRINT 'ДЛИНА ОТРЕЗКА D='S
130 INPUT 'ЗАДАЙТЕ M,N 'M,N:LETR=M+N
140 LET L=A:LETK=D:GOSUB 190:LET X=W
150 LET L=B:LETK=E:GOSUB 190:LET Y=W
160 LET L=C:LETK=F:GOSUB 190:LET Z=W
170 PRINT 'КООРДИНАТЫ ТОЧКИ ДЕЛЕНИЯ'
180 PRINT 'X='X, 'Y='Y, 'Z='Z:GOTO 130
190 LET W=(N*L+M*K)/R:RETURN:END

```

Пример. Для векторов $A_1(8; -3; 8)$ и $A_2(6; -1; 9)$ получим $Q=14,6712486^\circ$ и $D=3$. Задав $M=1$ и $N=4$, находим $X=7,6$, $Y=-2,6$ и $Z=8,2$.

Плоскость в пространстве декартовой системы координат, описываемая уравнением $AX + BY + CZ + D = 0$, имеет полярные параметры (см. рис. 3.5)

$$P = |D| / \sqrt{A^2 + B^2 + C^2}$$

$$\alpha = \arccos (\mp A / \sqrt{A^2 + B^2 + C^2}),$$

$$\beta = \arccos (\mp B / \sqrt{A^2 + B^2 + C^2}),$$

$$\gamma = \arccos (\mp C / \sqrt{A^2 + B^2 + C^2}),$$

где P — кратчайшее расстояние от точки $(0,0,0)$ до плоскости, α , β и γ — полярные углы (между перпендикуляром, опущенным на плоскость из точки $(0,0,0)$ и осями координат).

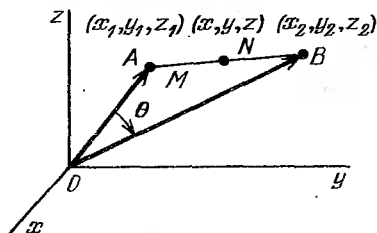


Рис. 3.4. Отрезок в пространстве, образованный двумя векторами, исходящими из точки $(0, 0, 0)$

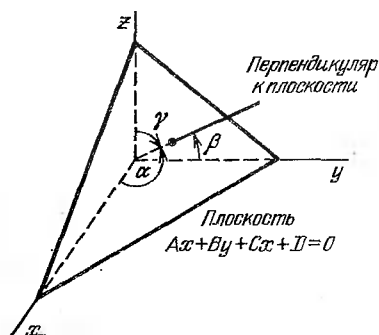


Рис. 3.5. Полярные параметры плоскости, расположенной в пространстве

Ориентированное расстояние от точки (X_1, Y_1, Z_1) до данной плоскости

$$L = \frac{AX_1 + BY_1 + CZ_1 + D}{\sqrt{A^2 + B^2 + C^2}}$$

Знак L указывает на расположение точки (X_1, Y_1, Z_1) : сверху или снизу плоскости.

Программа 3.68.

```
10 PRINT 'ПОЛЯРНЫЕ ПАРАМЕТРЫ ПЛОСКОСТИ A*X+B*Y+C*Z+D=0'  
20 PRINT ' И РАССТОЯНИЕ ОТ НЕЕ ДО ТОЧКИ M1(X1,Y1,Z1)'  
30 INPUT 'ВВЕДИТЕ ПАРАМЕТРЫ A, B, C, D 'A, B, C, D  
40 LETR=1: IF D>0 THEN LETR=-1  
50 LETM=SQR(A*A+B*B+C*C): LETA1=DEG(ACS(R*A/M))  
60 LETB1=DEG(ACS(R*B/M)): LETG1=DEG(ACS(R*C/M))  
70 PRINT 'УГЛЫ В ГРАДУСАХ': PRINT 'АЛЬФА='A1  
80 PRINT 'БЕТА='B1: PRINT 'ГАММА='G1  
90 INPUT 'ВВЕДИТЕ КООРДИНАТЫ ТОЧКИ X1, Y1, Z1 'X, Y, Z  
100 LETL=(A*X+B*Y+C*Z+D)/M  
110 PRINT 'РАССТОЯНИЕ ДО ТОЧКИ L='L: GOTO 90: END
```

Для $A=1$, $B=-2$, $C=2$ и $D=-3$ получаем $P=1$, $\alpha=70,52877937^\circ$, $\beta=131,8103149^\circ$ и $\gamma=48,1896851^\circ$. Для точки $(5,5,5)$ $L=0,666666667$.

АЛГОРИТМЫ И ПРОГРАММЫ РЕАЛИЗАЦИИ ОСНОВНЫХ ЧИСЛЕННЫХ МЕТОДОВ

§ 4.1. Решение систем линейных уравнений

Системы из n линейных уравнений вида

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2, \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (4.1)$$

решаются точными и итерационными методами. Точные методы дают точное решение за конечное число операций, если все они выполнены без погрешности. Число операций у итерационных методов зависит от заданной погрешности вычислений.

Метод Гаусса или *метод последовательного исключения неизвестных* основан на приведении матрицы коэффициентов a_{ij} к треугольному виду. При этом алгоритм решения системы (4.1) следующий.

1. С помощью двух циклов с управляющими переменными $i=1, 2, \dots, n$ и $j=1, 2, \dots, n$ организуем ввод коэффициентов a_{ij} и b_i , образующих массивы $A(I, J)$ и $B(I)$.

2. Проводим прямой ход исключения переменных путем преобразования коэффициентов (4.1) по формулам

$$a_{ij} = -a_{ij}/a_{ii}; \quad a_{jk} = a_{jk} + a_{ij}a_{ik}; \quad b_j = b_j + a_{ij}b_i,$$

где $i=1, 2, \dots, n-1$; $j=i+1, i+2, \dots, n$ и $k=i+1, i+2, \dots, i+n$. В конце этих преобразований получаем

$$x_n = b_n/a_{nn}.$$

3. Организуем обратный ход (последовательное нахождение $x_{n-1}, x_{n-2}, \dots, x_2, x_1$), проводя вычисления по формулам

$$h = b_i \text{ и } h = h - x_j a_{ji},$$

где $i=n-1, n-2, \dots, 2, 1$; $j=i+1, i+2, \dots, n$ и $x_i = h/a_{ii}$. В результате формируется массив $X(I)$ неизвестных x_1, x_2, \dots, x_n .

4. Организуем вывод массива $X(I)$.

Программа 4.1.

Пр.и.м.е.р. Для системы ($n=3$)

$$\begin{aligned} 4x_1 + 0,24x_2 - 0,08x_3 &= 8, \\ 0,09x_1 + 3x_2 - 0,15x_3 &= 9, \\ 0,04x_1 - 0,08x_2 + 4x_3 &= 20 \end{aligned} \quad (4.2)$$

находим $x_1 = 1,909198281$; $x_2 = 3,194964417$ и $x_3 = 5,044807306$. Если коэффициенты a_{ij}

близки к 0, может наступить аварийный останов ЭВМ, либо из-за погрешностей округления точность сильно ухудшается. Это является недостатком простейшей реализации метода Гаусса.

Метод Гаусса с выбором главного элемента заключается в том, что при прямом ходе производится выбор наибольшего по модулю (главного) элемента и перестановка строк или столбцов (см. рис. 4.1). Последнее исключает деление на 0; если матрица $A(I, J)$ содержит нулевые элементы, и повышает точность вычислений при наличии ошибок округления. Для ПЭВМ, ведущих вычисления с числами с плавающей запятой, достаточен выбор $A(I, I) \neq 0$.

Программа 4.2.

Пр.и.м.е.р. Для системы

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 4, \\ x_1 + x_2 - x_3 &= 1, \\ x_1 - 2x_2 + x_3 &= 3. \end{aligned} \quad (4.3)$$

получим $x_1 = 1,7$; $x_2 = -0,6$ и $x_3 = 0,1$.

Метод вращения [24] является разновидностью метода Гаусса, обладающей повышенной устойчивостью к «провалам» промежуточных вычислений. Этот метод обеспечивает приведение исходной системы (4.1) к системе с правой треугольной матрицей по следующему алгоритму.

1. Вводим a_{ij} и b_i .

2. Вычисляем

$$M_{ki} = \frac{a_{ki}}{\sqrt{a_{ii}^2 + a_{ki}^2}} \quad \text{и} \quad L_{ki} = -\frac{a_{ki}}{\sqrt{a_{ii}^2 + a_{ki}^2}},$$

причем если $a_{ii} = a_{ki} = 0$, то $M_{ki} = 1$ и $L_{ki} = 0$. Проводим преобразования системы по формулам

$$\begin{aligned} M_{ki}y_i - L_{ki}y_k &= M_{ki}b_i - L_{ki}b_k, \\ L_{ki}y_i - M_{ki}y_k &= L_{ki}b_i - M_{ki}b_k, \end{aligned}$$

где $i=1, 2, \dots, n+1$, $k=i+1, i+2, \dots, n$, y_i и y_k — левые части, а b_i и b_k — правые части i -го и k -го уравнений соответственно.

После $n(n-1)/2$ шагов приходим к системе с верхней треугольной матрицей.

3. Осуществляем обратный ход:

$$x_m = (b_m^{(1)} - a_{m, m+1}^{(1)} x_{m+1} - \dots - a_{m, n}^{(1)} x_n) / a_{mm}^{(1)}$$

где $m = n, n-1, \dots, 1$.

Программа 4.1.

```

05 PRINT 'РЕШЕНИЕ СИСТЕМЫ ИЗ N ЛИНЕЙНЫХ УРАВНЕНИЙ'
10 INPUT 'ЧИСЛО УРАВНЕНИЙ N=' N
20 DIM A(N,N), B(N), X(N)
30 FOR I=1 TO N: PRINT !2.0!' ВВОД КОЭФ. УРАВНЕНИЯ ' I
40 FOR J=1 TO N: INPUT A(I,J)
50 NEXT J: INPUT B(I): NEXT I
60 FOR I=1 TO N-1: FOR J=I+1 TO N
70 LET A(J,I)=-A(J,I)/A(I,I): FOR K=I+1 TO N
80 LET A(J,K)=A(J,K)+A(J,I)*A(I,K): NEXT K
90 LET B(J)=B(J)+A(J,I)*B(I): NEXT J: NEXT I
100 LET X(N)=B(N)/A(N,N)
110 FOR I=N-1 TO 1 STEP -1: LET H=B(I)
120 FOR J=I+1 TO N: LET H=H-X(J)*A(I,J): NEXT J
130 LET X(I)=H/A(I,I): NEXT I
140 PRINT 'КОРНИ СИСТЕМЫ УРАВНЕНИЙ'
150 FOR I=1 TO N: PRINT !2.0!' X(' I )=' : PRINT !E! X(I)
160 NEXT I: END

```

Программа 4.2.

```

10 PRINT 'РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДОМ'
20 PRINT ' ГАУССА С ВЫБОРОМ ГЛАВНОГО ЭЛЕМЕНТА'
30 INPUT 'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N= ' N
40 DIM A(N,N), B(N), C(N,N), G(N), X(N)
50 FOR I=1 TO N: FOR J=1 TO N
60 PRINT !2.0!' ВВЕДИТЕ A' I ', ' J '=' : INPUT A(I,J): NEXT J
70 PRINT 'ВВЕДИТЕ B' I '=' : INPUT B(I): NEXT I
80 GOSUB 100
85 FOR I=1 TO N
90 PRINT !2.0!' X' I '=' !F1.9! X(I): NEXT I: STOP
100 LET N1=N-1: FOR K=1 TO N1
110 IF ABS(A(K,K)) >> 0 GOTO 200
120 LET K1=K+1: FOR M=K1 TO N
130 IF ABS(A(M,K)) >> 0 GOTO 150
140 GOTO 165
150 FOR L=1 TO N: LET U=A(K,L): LET A(K,L)=A(M,L)
160 LET A(M,L)=U: NEXT L
165 NEXT M
170 LET U=B(K): LET B(K)=B(M): LET B(M)=U
200 LET G(K)=B(K)/A(K,K): LET K1=K+1
210 FOR I=K1 TO N: LET B(I)=B(I)-A(I,K)*G(K)
220 FOR J1=K TO N: LET J=N-J1+K: LET C(K,J)=A(K,J)/A(K,K)
225 LET A(I,J)=A(I,J)-A(I,K)*C(K,J)
230 NEXT J1: NEXT I: NEXT K
240 LET M=N: LET X(M)=B(M)/A(M,M)
250 LET M=M-1: LET S=0: FOR L=M TO N1
260 LET S=S+C(M,L+1)*X(L+1): NEXT L
270 LET X(M)=G(M)-S: IF M > 1 GOTO 250
280 RETURN: END

```

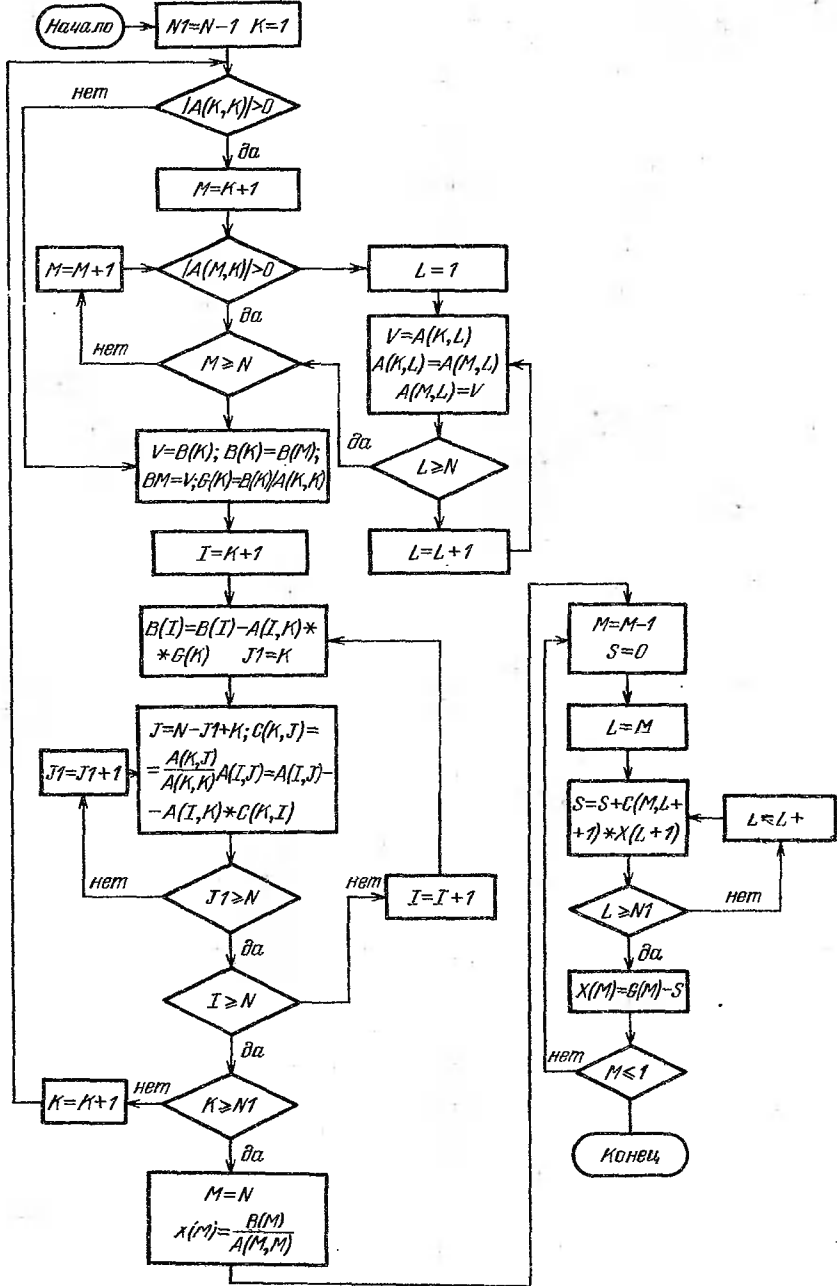


Рис. 4.1. Алгоритм реализации метода Гаусса с выбором главного элемента

```

10 PRINT 'РЕШЕНИЕ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ '
15 PRINT ' МЕТОДОМ ВРАЩЕНИЯ'
20 INPUT 'ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N=':N:LETM=0:DIM A(N,N)
25 FOR I=1 TO N:FOR J=1 TO N
30 PRINT!2.0!'ВВЕДИТЕ КОЭФФИЦИЕНТ A'I', 'J
40 INPUT A(I,J):IF J<N THEN 60
50 PRINT'ВВЕДИТЕ B'I:INPUT A(I,0)
60 NEXT J:NEXT I
70 FOR I=1 TO N-1:FOR K=I+1 TO N
80 IF A(I,I)<>0 THEN 110
90 IF A(K,I)<>0 THEN 110
100 LETM=1:LETL=0:GOTO 130
110 LETM=SGR(A(I,I)^2+A(K,I)^2)
120 LETL=-A(K,I)/M:LETM=A(I,I)/M
130 FOR J=1 TO N:LETR=M*A(I,J)-L*A(K,J)
140 LETA(K,J)=L*A(I,J)+M*A(K,J)
150 LETA(I,J)=R:NEXT J
160 LETR=M*A(I,0)-L*A(K,0)
170 LETA(K,0)=L*A(I,0)+M*A(K,0)
180 LETA(I,0)=R:NEXT K: NEXT I
190 FOR I=N TO 1 STEP -1:LETM=0
200 FOR K=0 TO N-I-1
210 LETM=M+A(0,N-K)*A(I,N-K):NEXT K
220 LETA(0,I)=(A(I,0)-M)/A(I,I)
230 PRINT!2.0!'КОРЕНЬ X'I '=:!F1.9!A(0,I)
240 NEXT I:END
    
```

Для проверки программы можно решить уравнение (4.3). Результат совпадает с указанным выше.

Метод простых итераций, описанный в § 4.4 для систем нелинейных уравнений, применим и для решения систем линейных уравнений. Однако его сходимость гарантируется, если значения диагональных элементов матрицы $A(I, J)$ превосходят остальные, что снижает применимость метода простых итераций (хотя любые системы линейных уравнений можно свести к нужному для сходимости виду с помощью преобразований, описанных в [7]). Сказанное относится и к методу Зейделя.

Программы реализации этих и некоторых других методов частного применения даны в Приложении 5.

Отметим, что итерационные методы обладают свойством самоисправления ошибок в ходе вычислений и могут применяться в особых случаях, например, когда матрица коэффициентов a_{ij} сильно разрежена, т. е. содержит много нулей.

Метод минимизации заключается в поиске минимума целевой функции

$$F(x_1, x_2, \dots, x_n) = |f_1(x_1, x_2, \dots, x_n)| + \dots + |f_2(x_1, x_2, \dots, x_n)|,$$

компоненты которой формируются из уравнений решаемой системы

$$f_1(x_1, x_2, \dots, x_n) = a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n - b_1,$$

$$f_2(x_1, x_2, \dots, x_n) = a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n - b_2,$$

$$f_n(x_1, x_2, \dots, x_n) = a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n - b_n.$$

Если x_1, x_2, \dots, x_n — решение (4.1), то функ-

ция $F(x_1, x_2, \dots, x_n) = 0$. Для реализации этого метода могут использоваться программы поиска минимума функции ряда переменных, описанные в § 5.6.

§ 4.2. Интерполяция и экстраполяция

Интерполяция функции $y(x)$ одной переменной x , заданной $(n+1)$ узлами $y_i(x_i)$, где $i=0, 1, 2, \dots, n$, заключается в нахождении значений y по значениям x , находящихся в промежутках между узлами x_i . При интерполяции функция $y(x)$ заменяется интерполяционным полиномом $P(x)$, значения которого $P(x_i)$ в узлах точно совпадают с $y(x_i)$. Значение n задает степень полинома $P(x)$.

Формулы Лагранжа для интерполяции при равномерном расположении узлов обеспечивают наименьшее время интерполяции, не требуют обновления ввода y_i и x_0 для вычисления каждого $y(x)$ и позволяют вычислять y_i в узлах x_i (например, для контроля правильности вычислений). В этих формулах индексом 0 обозначается центральный узел. Для $n+1=2 \div 6$ формулы Лагранжа имеют вид [36]

$$y(x)_2 = (1-p)y_0 + py_1 + 0,125h^2y''(\xi);$$

$$y(x)_3 = \frac{p(p-1)}{2}y_{-1} + (1-p^2)y_0 + \frac{p(p+1)}{2}y_1 + 0,065h^3y'''(\xi);$$

$$y(x)_4 = \frac{p(p-1)(p-2)}{6}y_{-1} + \frac{(p^2-1)(p-2)}{2}y_0 - \frac{p(p+1)(p-2)}{2}y_1 + \frac{p(p^2-1)}{2}y_2 + R(p),$$

$$R(p) = \begin{cases} 0,024h^4 y^{IV}(\xi) & \text{при } 0 < p < 1, \\ 0,042h^4 y^{IV}(\xi) & \text{при } -1 < p < 0, \\ & 1 < p < 2; \end{cases}$$

$$y(x)_6 = \frac{(p^2-1)p(p-2)}{24} y_{-2} + \frac{(p-1)p(p^2-4)}{6} y_{-1} + \frac{(p^2-1)(p^2-4)}{4} y_0 + \frac{(p+1)p(p^2-4)}{6} y_1 + \frac{(p^2-1)p(p+2)}{24} y_2 + R(p).$$

где

$$R(p) = \begin{cases} 0,012h^5 y^V(\xi) & \text{при } |p| < 1, \\ 0,031h^5 y^V(\xi) & \text{при } 1 < |p| < 2; \end{cases}$$

$$y(x)_6 = \frac{p(p^2-1)(p-2)(p-3)}{120} y_{-2} + \frac{p(p-1)(p^2-4)(p-3)}{24} y_{-1} - \frac{(p^2-1)(p^2-4)(p-3)}{12} y_0 + \frac{p(p+1)(p^2-4)(p-3)}{12} y_1 - \frac{p(p^2-1)(p+2)(p-3)}{24} y_2 + \frac{p(p^2-1)(p^2-4)}{120} y_3 + R(p).$$

Программа 4.4.

```

10 PRINT 'ИНТЕРПОЛЯЦИЯ ПРИ N=CONST ПО ФОРМУЛАМ ЛАГРАНЖА'
20 INPUT 'ЗАДАЙТЕ СТЕПЕНЬ ПОЛИНОМА M=1-5 M=':M:LETN=M+1
30 INPUT 'ВВЕДИТЕ X0,H 'Z,H: IF N=3 THEN 100
40 IF N=4 THEN 140
50 IF N=5 THEN 190
60 IF N=6 THEN 250
70 INPUT 'ВВЕДИТЕ Y0,Y1 'A,B
80 INPUT 'ВВЕДИТЕ X=X:LETP=(X-Z)/H
90 LETY=(1-P)*A+P*B:PRINT 'Y='Y:GOTO 80
100 INPUT 'ВВЕДИТЕ Y-1,Y0,Y1 'A,B,C
110 INPUT 'ВВЕДИТЕ X=X:LETP=(X-Z)/H
120 LETY=P*(P-1)*A/2+(1-P)*B+P*(P+1)*C/2
130 PRINT 'Y='Y:GOTO 110
140 INPUT 'ВВЕДИТЕ Y-1,Y0,Y1,Y2 'A,B,C,D
150 INPUT 'ВВЕДИТЕ X=X:LETP=(X-Z)/H:LETE=(P-2)/2
160 LETY=-P*(P-1)*E*A/3+(P)*B+(P-1)*E*B
170 LETY=Y-P*(P+1)*E*C+P*(P-1)*E*D/6
180 PRINT 'Y='Y:GOTO 150
190 INPUT 'ВВЕДИТЕ Y-2,Y-1,Y0,Y1,Y2 'A,B,C,D,E
200 INPUT 'ВВЕДИТЕ X=X:LETP=(X-Z)/H
210 LETF=(P-1)/2:LETK=(P-4)/2
220 LETY=F*(P-2)*A/12-(P-1)*B*(P-1)*E/3+F*(P-1)*E
230 LETY=Y-(P+1)*B*(P-1)*E/12+F*(P+2)*E/12
240 PRINT 'Y='Y:GOTO 200
250 INPUT 'ВВЕДИТЕ Y-2,Y-1,Y0,Y1,Y2,Y3 'A,B,C,D,E,F
260 INPUT 'ВВЕДИТЕ X=X:LETP=(X-Z)/H:LETM=P-1
270 LETI=M/24:LETJ=-P*(P-3):LETK=M-3:LETL=K*(P-3)/12
280 LETY=J*(P-2)*A/5+L*(P-1)*B/2-M*(P-1)*E/3
290 LETY=Y+P*(P+1)*D+J*(P+2)*E+P*(P-1)*F/5
300 PRINT 'Y='Y:GOTO 260:END
    
```

$$R(p) = \begin{cases} 0,0049h^6 y^{VI}(\xi) & \text{при } 0 < p < 1, \\ 0,0071h^6 y^{VI}(\xi) & \text{при } -1 < p < 0, 1 < p < 2, \\ 0,024 h^6 y^{VI}(\xi) & \text{при } -2 < p < -1, 2 < p < 3. \end{cases}$$

В этих формулах
 $x = x_0 + px, p = (x - x_0)/h,$

где h — шаг расположения узлов, индекс $y(x)$ соответствует числу узлов $(h+1)$, $y^{n+1}(\xi)$ — максимальное значение производной $y(x)$ для точки $x = \xi$, лежащей в пределах интерполяции. Последний член формулы (он в программах не вычисляется) характеризует погрешность интерполяции.

Пример (к программе 4.4). Для контроля этой программы выполнен интерполяцию функции Струве $H_0(x) = y(x)$ [36], заданной таблицей, при $x_0 = 1$ и $h = 0,2$: $y_{-2} = 0,36699114$; $y_{-1} = 0,4739944$; $y_0 = 0,5686566$; $y_1 = 0,648855$; $y_2 = 0,7117925$ и $y_3 = 0,7570255$.

1. При линейной интерполяции ($M=1$) для $x=1,1$ получим $y(1,1) = 0,6087558$.
2. При квадратичной интерполяции ($M=2$) получим $y(0,9) = 0,5231716$; $y(1,1) = 0,6104494$.
3. При кубической интерполяции ($M=3$) получим $y(0,9) = 0,5230539688$; $y(1,1) = 0,6105670313$.
4. При $M=4$ получим $y(0,9) = 0,5230357695$; $y(1,1) = 0,6105779508$.
5. При $M=5$, получим $y(0,9) = 0,5230350945$; $y(1,1) = 0,6105786258$.

Точные значения $H_0(x): H_0(0,9) = 0,523035$ и $H_0(1,1) = 0,6105787$.

Существует ряд специальных видов полинома $P(x)$ (Ньютона, Эверетта и др.) [4, 5, 7, 18, 24, 30, 38]. Однако следует помнить, что полином $P(x)$, имеющий $P(x_i) = y(x_i)$, является всегда единственным. Поэтому при пренебрежении погрешностью вычислений ЭВМ все виды интерполяции должны давать одинаковый результат. Это характерно для современных ПЭВМ, оперирующих числами с плавающей запятой и имеющих нередко скрытые разряды для повышения точности вычислений.

Интерполяция по методу Эйткина заключается в вычислении $y(x)$ при произвольно расположенных узлах без явного построения интерполяционного полинома. Последнее достигается путем последовательного применения формул линейной интерполяции:

$$y(x, x_0, x_1) = \frac{1}{x_1 - x_0} \left| \begin{array}{c} y_0 \ x_0 - x \\ y_1 \ x_1 - x \end{array} \right|,$$

$$y(x, x_0, x_2) = \frac{1}{x_2 - x_0} \left| \begin{array}{c} y_0 \ x_0 - x \\ y_2 \ x_2 - x \end{array} \right|,$$

$$y(x, x_0, x_1, x_2) = \frac{1}{x_2 - x_1} \left| \begin{array}{c} y(x, x_0, x_1) \ x_1 - x \\ y(x, x_0, x_2) \ x_2 - x \end{array} \right|$$

и т. д. В приведенной ниже программе с $M=2 \div 6$ для ускорения счета интерполяция ведется прямо по приведенным формулам.

Программа 4.5.

```

05 PRINT 'ИНТЕРПОЛЯЦИЯ ТАБЛИЦ С ЧИСЛОМ УЗЛОВ ДО ШЕСТИ'
10 INPUT 'ЧИСЛО УЗЛОВ ДО ШЕСТИ M=' M
20 LET B2=0: LET B3=0: LET B4=0: LET B5=0: LET B6=0: LET X3=0: LET X4=0: LET X5=0
30 INPUT 'ВВЕДИТЕ X0, Y0' X0, Y0
40 INPUT 'ВВЕДИТЕ X1, Y1' X1, Y1
50 LET B0=Y0: LET Q1=(Y1-Y0)/(X1-X0): LET B1=Q1
60 IF M=2 THEN 210
70 INPUT 'ВВЕДИТЕ X2, Y2' X2, Y2
80 LET Q2=(Y2-Y0)/(X2-X0): LET R2=(Q2-Q1)/(X2-X1)
90 LET B2=R2: IF M=3 THEN 210
100 INPUT 'ВВЕДИТЕ X3, Y3' X3, Y3
110 LET Q3=(Y3-Y0)/(X3-X0): LET R3=(Q3-Q1)/(X3-X1)
120 LET S3=(R3-R2)/(X3-X2): LET B3=S3: IF M=4 THEN 210
130 INPUT 'ВВЕДИТЕ X4, Y4' X4, Y4
140 LET Q4=(Y4-Y0)/(X4-X0): LET R4=(Q4-Q1)/(X4-X1)
150 LET S4=(R4-R2)/(X4-X2): LET L4=(S4-S3)/(X4-X3)
160 LET B4=L4: IF M=5 THEN 210
170 INPUT 'ВВЕДИТЕ X5, Y5' X5, Y5
180 LET Q5=(Y5-Y0)/(X5-X0): LET R5=(Q5-Q1)/(X5-X1)
190 LET S5=(R5-R2)/(X5-X2): LET L5=(S5-S3)/(X5-X3)
200 LET B5=(L5-L4)/(X5-X4)
210 INPUT 'ВВЕДИТЕ X=' X
220 PRINT 'Y=' (((B5*(X-X4)+B4)*(X-X3)+B3)*(X-X2)+B2)*(X-X1)+B1)*(X-X0)+B0
230 GOTO 210: END
    
```

Пример. Для контроля этой программы проведем интерполяцию гамма-функции $\Gamma(x) = y(x)$, заданной $M=6$ значениями x_i и y_i [36]: $x_0=1,5; y_0=0,8862269255; x_1=1,51; y_1=0,886591685; x_2=1,525; y_2=0,8872930231; x_3=1,54; y_3=0,8881776586; x_4=1,54; y_4=0,889639199; x_5=1,59$ и $y_5=0,8924282141$. Тогда для $x=1,55$ получим

$y(1,55) = 0,8888683478$ (все цифры результата верные).

Интерполяция полиномом Лагранжа при произвольном расположении узлов в общем случае сводится к вычислению $y(x) = L_n(x)$ с помощью интерполяционного полинома, имеющего вид (рис. 4.2)

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

Программа 4.6.

Пример. Используя данные примера к программе 4.5, получим $y(1,55) = 0,888868348$.

Интерполяция с одновременным получением коэффициентов полинома

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

может выполняться с применением интерполяционной формулы Ньютона [4, 24]:

$$P_{n-1}(x) = y_1 + (x-x_1)f(x_1; x_2) + (x-x_1)(x-x_2)f(x_1; x_2; x_3) + (x-x_1)(x-x_2)(x-x_3)f(x_1; x_2; x_3; x_4) + \dots + (x-x_{n-1})f(x_1; x_2; \dots; x_n) = \sum_{i=0}^{n-1} A_k \Phi_i;$$

где $A_0 = y_1, A_k = f(x_1; x_2; \dots, x_{k+1})$ — раздельные разности K -го порядка ($K=1, 2,$

$3, \dots, n-1$) и $\Phi_0 = 1$;

$$\Phi_1 = \prod_{j=1}^i (x-x_j) = \sum_{j=0}^i \Phi_j x^{i-j}; \Phi_k = (-1)^k \prod_{j=1}^k x_j;$$

$$\Phi_{k-i} = \frac{1}{i!} \sum_{m=1}^i (-1)^{m+1} \Phi_{k-i+m} \sum_{p=1}^k (-1/x_p)^m;$$

$$\Phi_0 = 1; i=1, 2, \dots, n-1; k=1, 2, \dots, i.$$

```

10 PRINT 'ИНТЕРПОЛЯЦИЯ ПО ЛАГРАНЖУ ДЛЯ N+1 УЗЛОВ'
20 INPUT 'ВВЕДИТЕ N=':N:DIM A(N):DIM B(N)
30 FOR I=0 TO N:PRINT!3.0!'ВВЕДИТЕ X' I
40 INPUT A(I):PRINT'ВВЕДИТЕ Y' I
50 INPUT B(I):NEXT I
60 INPUT'ВВЕДИТЕ X=':X:LETS=0
70 IF X=A(0) THEN PRINT!1.9!'Y(X)='B(0):GOTO 60
80 FOR J=1 TO N:LETC=1:FOR I=1 TO N
90 LETD=A(J)-A(I):IF I=J THEN LETD=X-A(J)
100 IF D=0 THEN PRINT!1.9!'Y(X)='B(I):GOTO 60
110 LETC=C*(X-A(I))/D:NEXT I
120 LETS=S+C*B(J):NEXT J
130 PRINT!1.9!'Y('X')='S':GOTO 60:END
    
```

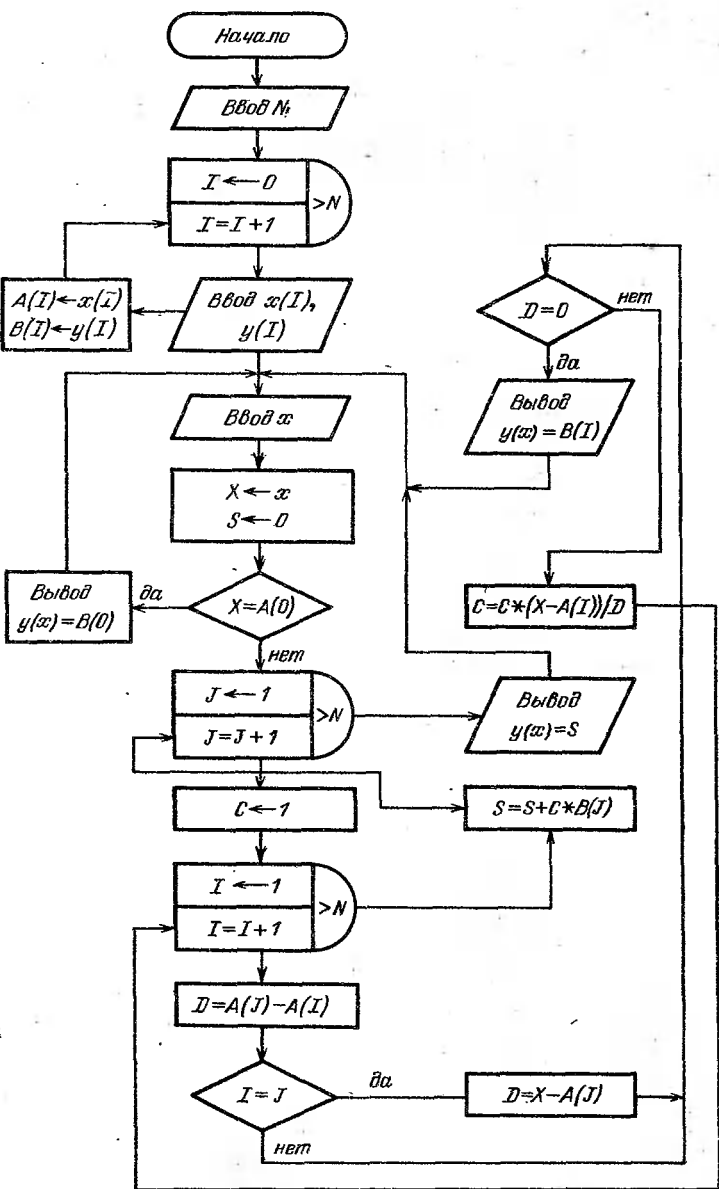


Рис. 4.2. Алгоритм интерполяции полиномом Лагранжа

```

10 PRINT 'ПОСТРОЕНИЕ ИНТЕРПОЛЯЦИОННОГО ПОЛИНОМА НЬЮТОНА'
20 PRINT 'И ИНТЕРПОЛЯЦИИ ПРИ ПРОИЗВОЛЬНО РАСПОЛОЖЕННЫХ УЗЛАХ'
30 INPUT 'ЗАДАЙТЕ ЧИСЛО УЗЛОВ N=';N:DIM A(N),F(N),X(N),Y(N)
40 FOR I=1 TO N:PRINT!2.0!'ВВЕДИТЕ X'I,'Y'I
50 INPUT X(I),Y(I):NEXT I:LETA(1)=1:LETF(N)=Y(1)
60 FOR I=1 TO N-1:LETF(I)=0:NEXT I
70 FOR K=1 TO N-1:FOR I=1 TO N-K
80 LETY(I)=(Y(I+1)-Y(I))/(X(I+K)-X(I)):NEXT I
90 LETR=1:IF K/2-INT(K/2)<>0 THEN LETR=-1
100 LETP=1:FOR J=1 TO K:LETR=P*X(J):NEXT J
110 LEYA(K+1)=R*P:IF K=1 THEN 170
120 FOR L=1 TO K:LETW=0:FOR M=1 TO L
130 LETR=1:IF K/2-INT(K/2)<>0 THEN LETR=-1
140 LETS=0:FOR P=1 TO K:LETS=S+R*(1/X(P))^M:NEXT P
150 LETW=W+(-R)*A(K+1+M-L)*S:NEXT M
160 LETA(K-L+1)=W/L:NEXT L
170 FOR J=N TO N-K STEP -1
180 LETF(J)=F(J)+A(J-N+K+1)*Y(1):NEXT J:NEXT K
190 PRINT 'КОЭФФИЦИЕНТЫ СТЕПЕННОГО МНОГОЧЛЕНА'
200 FOR I=1 TO N:PRINT!2.0!'A'N-I!F1.9!'='F(I):NEXT I
210 INPUT 'ВВЕДИТЕ ЗНАЧЕНИЕ X=';X:LETS=F(1):FOR I=1 TO N-1
220 LETS=S*X+F(I+1):NEXT I
230 PRINT!F1.9!'Y(X)=';S:GOTO 210:END

```

Пример. Пусть надо построить интерполяционный полином для янтеграла вероятности $y(x) = \Phi_0(x) = (1/\sqrt{2\pi}) \int_0^x \exp(-t^2) dt$,

заданного пятью узлами $y_1(2,2) = 0,4860966$; $y_2(2,3) = 0,4892759$; $y_3(2,4) = 0,4918025$; $y_4(2,5) = 0,4937903$ и $y_5(2,6) = 0,4953388$. Введя эти значения, найдем: $a_0 = -0,1706447$; $a_1 = 0,805165666$; $a_2 = -0,3712775$; $a_3 = -0,07783333333$ и $a_4 = -0,0625$. Далее можно вычислить $y(x)$ для заданного x (в том числе и в узлах интерполяции).

Экстраполяция — получение значений $y(x)$ при x , не принадлежащем отрезку $[x_0, x_n]$ или $[x_1, x_{n+1}]$, также осуществляется по описанным выше программам, но с существенно большей погрешностью. Для гладких $y(x)$ экстраполяция целесообразна при x , выходящих за указанные пределы не более чем на $h/2$.

Обратная интерполяция — процесс нахождения значений x по заданным значениям y . Она может выполняться по любой программе интерполяции с произвольно расположенными узлами. При этом вместо значений x_i вводятся значения y_i , а вместо y_i — значения x_i .

Многоинтервальная интерполяция заключается в интерполяции $y_i(x_i)$ в ряде частичных интервалов (ограниченных двумя узлами или группой узлов) отдельными полиномами невысокой степени. Такая интерполяция может применяться при широком общем отрезке $[a, b]$, когда обычная интерполяция полиномом высокой степени дает большую погрешность и ведет к большому времени вычислений. Заметим также, что по виду полинома и значениям его коэффициентов трудно судить о виде зависимости $y(x)$.

Многоинтервальная кусочно-линейная интерполяция при равномерном расположении узлов сводится к заданию начального значения $x_0 = a$, шага h (расстояния между узлами), номера n последнего узла и $(n+1)$

ординат y_0, y_1, \dots, y_n , после чего вычисление $y(x)$ при заданном x выполняется по формулам

$$i = \text{int}((x-a)/h),$$

$$y(x) = y_i + (y_{i+1} - y_i)(x - ih - a)/h.$$

На примере многоинтервальной кусочно-линейной интерполяции отчетливо видны общие свойства многоинтервальной интерполяции: степень интерполирующего полинома (в данном случае 1) не зависит от числа узлов; с ростом последнего погрешность интерполяции монотонно стремится к нулю; для любого интервала вычисление $y(x)$ производится по одним и тем же относительно простым (из-за малой степени полинома) формулам, поэтому время вычислений $y(x)$ при заданных x мало; массив y_i несет наглядную информацию о виде зависимости $y(x)$.

Задание асимптотического поведения $y(x)$ при $x < x_0 = a$ и $x > b = a + nh$ заключается в линейной экстраполяции, т. е. вычислении $y(x)$ за пределами отрезка $[a, b]$ по формулам

$$y(x) = y_0 + (x-a)(y_1 - y_0)/h \text{ при } x < a,$$

$$y(x) = y_n + (x-b)(y_n - y_{n-1})/h \text{ при } x > b.$$

Такое задание не всегда строго, но позволяет избежать грубых искажений асимптотического поведения экстраполирующей функции $y(x)$ за пределами отрезка $[a, b]$, которые нередко наблюдаются при обычной полиномиальной экстраполяции.

Многоинтервальная квадратичная интерполяция заключается в задании четного числа парных интервалов (n — четное число) с вычислением $y(x)$ при заданном x по формулам ($x_0 = a$)

$$i = \text{int}((x-a)/2h) + 1, p = (x-a-ih)$$

и

$$y(x) = \frac{p(p-1)}{2} y_{i-1} + (1-p^2) y_i + \frac{p(p+1)}{2} y_{i+1}.$$

Выражение для $y(x)$ — это интерполяционная формула Лагранжа для трех ординат, применяемая для каждого частичного интервала.

Асимптотическое поведение $y(x)$ задается с помощью линейной экстраполяции с вычислением первой производной в концах общего отрезка $[a, b]$ по формулам численного дифференцирования для трех точек:

$$y'(x) = y_0 + (x-a)(4y_1 - y_2 - 3y_0)/2h \quad \text{при } x \leq a,$$

$$y'(x) = y_n + (x-b)(3y_n + y_{n-2} - 4y_{n-1})/2h$$

при $x \geq b$.

Сплайн-интерполяция есть специальный вид многонтервальной интерполяции, при котором интерполирующий полином обеспечивает не только равенство $y(x)$ значениям y_i в узлах, но и непрерывность заданного числа первых производных на границах частичных интервалов. В общем случае сплайн задается глобальным способом, т. е. с использованием всех узлов при любом их расположении (см. § П5.11 Приложения 5). Ниже рассматривается задание кубического сплайна локальным способом, которое реализуется сравнительно просто и требует существенно меньшего объема памяти ЭВМ, чем при глобальном способе задания.

Кубический сплайн, заданный локально, — это интерполирующая функция в виде полинома третьей степени, вычисляемая по формулам [6],

$$i = \text{int}((x-a)/h),$$

$$y(x) = \frac{(x_{i+1}-x)^2(2(x-x_i)+h)}{h^3} y_i + \frac{(x-x_i)^2(2(x_{i+1}-x)+h)}{h^3} y_{i+1} + \frac{(x_{i+1}-x)^2(x-x_i)}{h^2} m_i - \frac{(x-x_i)^2(x-x_{i+1})}{h^2} m_{i+1},$$

где m_i и m_{i+1} — первые производные $y(x)$. Производные локального сплайна могут задаваться двумя способами.

Способ 1. Производные m_i и m_{i+1} вычисляются с помощью формул численного дифференцирования по трем точкам:

$$m_i = (y_{i+1} - y_{i-1})/2h \quad \text{для } i=1, 2, \dots, n-1,$$

$$m_0 = (4y_1 - y_2 - 3y_0)/2h \quad \text{для } i=0,$$

$$m_n = (3y_n + y_{n-2} - 4y_{n-1})/2h \quad \text{для } i=n.$$

Способ удобен тем, что для задания сплайна требуется вводить лишь ординаты y_i (значения m_i вычисляются программой). Для уменьшения времени многократных вычислений $y(x)$ желательно предварительно вычислить массив m_i и хранить его в памяти ЭВМ.

Способ 2. Значения m_i (вычисленные отдельно или полученные из графика как наклоны его в узлах) задаются непосредственно в виде массива m_i .

Задание асимптотического поведения $y(x)$ за пределами отрезка $[a, b]$ осуществляется так же, как и при многонтервальной квадратичной интерполяции. У рассмотренного локально заданного сплайна непрерывны лишь нулевая и первая производные (у глобально заданных кубических сплайнов непрерывна также и вторая производная). Кусочно-линейная и многонтервальная квадратичная интерполяция рассматриваются как дефектные сплайны, у которых обеспечивается непрерывность только нулевой производной. Все эти виды интерполяции реализуются программой 4.8. После задания нужного вида интерполяции вычисления $y(x)$ выполняются подпрограммами: кусочно-линейная интерполяция — экстраполяция со строки 1000, квадратичная — со строки 2000 и кубическая — со строки 3000. Эти подпрограммы могут использоваться для задания (аппроксимаций) нелинейных зависимостей $y(x)$ в дополнительных программах, вписываемых пользователем в программу 4.8.

Программа 4.8.

```

10 PRINT 'АППРОКСИМАЦИЯ ИНТЕРПОЛЯЦИЯ И ЭКСТРАПОЛЯЦИЯ'
20 PRINT 'СПЛАЙНАМИ ПРИ ЛОКАЛЬНОМ ИХ ЗАДАНИИ'
30 INPUT 'ВВЕДИТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X0=';A
40 INPUT 'ВВЕДИТЕ ШАГ H=';H
50 INPUT 'ВВЕДИТЕ НОМЕР ПОСЛЕДНЕГО УЗЛА N=';N:DIM Y(N)
60 FOR I=0 TO N:PRINT!3.0!'ВВЕДИТЕ Y'I
70 INPUT Y(I):NEXT I
80 INPUT 'ВВЕДИТЕ ПОРЯДОК 1,2 ИЛИ 3 ПОЛИНОМА'P
90 IF P=2 THEN 130
100 IF P=3 THEN 150
110 INPUT 'ВВЕДИТЕ X=';X:GOSUB 1000
120 PRINT!F1.9!'Y(X)=';W:GOTO 110
130 INPUT 'ВВЕДИТЕ X=';X:GOSUB 2000
140 PRINT!F1.9!'Y(X)=';W:GOTO 130
150 PRINT 'ЗАДАЙТЕ КОД 0-DY/DX НЕ ЗАДАЕТСЯ':DIM M(N)
160 INPUT 'ЗАДАЙТЕ КОД 1-DY/DX ЗАДАЕТСЯ'K
170 IF K=0 THEN 200
180 FOR I=0 TO N:PRINT!3.0!'ВВЕДИТЕ DY/DX'I
190 INPUT M(I):NEXT I:GOTO 230
200 LETM(0)=(4*Y(1)-Y(2)-3*Y(0))/2/H
210 LETM(N)=(3*Y(N)+Y(N-2)-4*Y(N-1))/2/H
220 FOR I=1 TO N-1:LETM(I)=(Y(I+1)-Y(I-1))/2/H:NEXT I
230 INPUT 'ВВЕДИТЕ X=';X:GOSUB 3000

```



```

240 PRINT IF 1.91*(Y(X)-W)/H:GOTO 230
1000 REM'ВЫЧИСЛЕНИЕ Y(X)=W ПРИ СТЕПЕНИ ПОЛИНОМА 1'
1010 IF X<=A THEN 1050
1020 LET B=A+H*N:IF X>=B THEN 1060
1030 LET I=INT((X-A)/H):LET W=Y(I)
1040 LET W=(Y(I+1)-Y(I))*((X-I)*H-A)/H:RETURN
1050 LET W=Y(0)+(X-A)*(Y(1)-Y(0))/H:RETURN
1060 LET W=Y(N)+(X-B)*(Y(N)-Y(N-1))/H:RETURN
2000 REM'ВЫЧИСЛЕНИЕ Y(X)=W ПРИ СТЕПЕНИ ПОЛИНОМА 2'
2010 IF X<=A THEN 2060
2020 LET B=A+H*N:IF X>=B THEN 2070
2030 LET I=1+2*INT((X-A)/2/H):LETP=(X-A-I)*H/H
2040 LET W=P*(P-1)*Y(I-1)/2+(1-P)*Y(I)
2050 LET W=W+P*(P+1)*Y(I+1)/2:RETURN
2060 LET W=Y(0)+(X-A)*(4*Y(1)-Y(2)-3*Y(0))/2/H:RETURN
2070 LET W=Y(N)+(X-B)*(3*Y(N)+Y(N-2)-4*Y(N-1))/2/H:RETURN
3000 REM'ВЫЧИСЛЕНИЕ Y(X)=W ПРИ СТЕПЕНИ ПОЛИНОМА 3'
3010 IF X<=A THEN 3080
3020 LET B=A+H*N:IF X>=B THEN 3090
3030 LET I=INT((X-A)/H):LET B=A+H*I
3040 LET C=B+H:LET D=X-C:LETE=X-B
3050 LET W=D*D*(E+E+H)*Y(I)+E*E*(H-D-D)*Y(I+1)
3060 LET W=W/H+D*D*E*M(I)
3070 LET W=(W+E*E*D*M(I+1))/H/H:RETURN
3080 LET W=Y(0)+(X-A)*M(0):RETURN
3090 LET W=Y(N)+(X-B)*M(N):RETURN:END

```

Пример. Провести интерполяцию (аппроксимацию) N -образной вольт-амперной характеристики туннельного диода, заданной в виде

влияние на ход $y(x)$ по всей кривой, что качественно неверно описывает физические явления, лежащие в основе нелинейности интерполируемой функции.

$x_i = U, В$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
$y_i = I, мА$	0	10	4	2	1,5	2	5	9	13

Задаем $x_0 = 0$, $h = 0,1$ и $n = 8$. Введя ординаты y_i для кусочно-линейной интерполяции — экстраполяции, будем иметь $y(0,1) = 10$, $y(-0,1) = -10$, $y(0,05) = 5$, $y(0,8) = 13$ и т. д. Для многоинтервальной квадратичной интерполяции — экстраполяции $y(0,1) = 10$, $y(-0,1) = -18$, $y(0,25) = 2,8125$, $y(1) = 21$ и т. д. Для кубической сплайн-интерполяции и экстраполяции $y(0,1) = 10$; $y(0,25) = 2,65625$, $y(-0,1) = -18$, $y(1) = 21$ и т. д.

Из этого примера видно, что для задания сплайна, аппроксимирующего некоторую зависимость $y(x_i)$, достаточно задать лишь определенное число ее ординат y_i . Массив значений y_i сохраняется в памяти ЭВМ и в любой момент может быть использован для восстановления вида аппроксимируемой зависимости, например, для вывода по точкам ее графика.

Достоинством локально заданных сплайнов является описание свойств зависимости $y_i(x_i)$ на каждом частичном интервале независимо от ее свойств на других интервалах. Подобное поведение нередко встречается на практике, например, восходящие туннельная и диффузионная ветви N -образной вольт-амперной характеристики туннельного диода обусловлены совершенно различными физическими явлениями. Отметим, что у обычной полиномиальной аппроксимации этого свойства нет и изменение $y(x)$ вблизи какого-либо узла оказывает

более точной, но и более сложной, является аппроксимация сплайнами при глобальном их задании. В этом случае коэффициенты m_i вычисляются по всем узлам $y_i(x_i)$. Этот вид аппроксимации рассматривается в § П5.11.

Интерполяция функции двух переменных x и y по трем точкам (см. рис. 4.3, а) выполняется по формулам

$$f(x, y) = (1-p-q)f_{00} + pf_{10} + qf_{01},$$

где $p = (x-x_0)/h$ и $q = (y-y_0)/l$. Здесь h — шаг изменения x , а l — шаг изменения y . Несколько лучшие результаты дает интерполяция по четырем точкам (рис. 4.3, б):

$$f(x, y) = (1-p)(1-q)f_{00} + p(1-q)f_{10} + q(1-p)f_{01} + pqf_{11}.$$

Ошибка при этом пропорциональна h^2 .

При интерполяции по шести точкам (рис. 4.3, в)

$$f(x, y) = \frac{q(q-1)}{2}f_{0-1} + \frac{p(p-1)}{2}f_{-10} + (1+pq-p^2-q^2)f_{00} + \frac{p(p-2q+1)}{2}f_{10} + \frac{q(q-2p+1)}{2}f_{01} + pqf_{11}$$

погрешность пропорциональна h^3 .

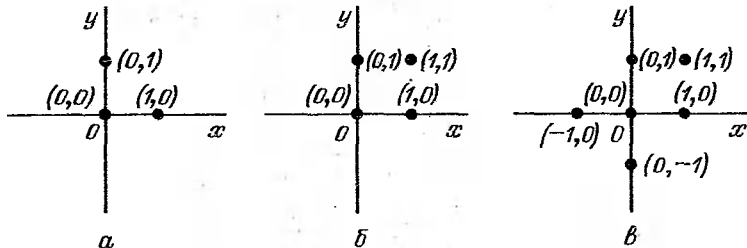


Рис. 4.3. Расположение равноотстоящих узлов при интерполяции функции двух переменных по 3(а), 4(б) и 7(в) точкам

Программа 4.9.

```

10 PRINT 'ИНТЕРПОЛЯЦИЯ ФУНКЦИИ ДВУХ ПЕРЕМЕННЫХ'
20 PRINT 'F(X,Y) ПО 3,4 И 6 ТОЧКАМ'
30 INPUT 'ЗАДАЙТЕ ШАГ X H='
40 INPUT 'ЗАДАЙТЕ ШАГ Y L='
50 INPUT 'ЗАДАЙТЕ ЧИСЛО ТОЧЕК 3,4 ИЛИ 6 N='
60 IF N=4 THEN 90
70 IF N=6 THEN 100
80 INPUT 'ВВЕДИТЕ F00,F10,F01 'A,B,C:GOTO 110
90 INPUT 'ВВЕДИТЕ F00,F10,F01,F11 'A,B,C,D:GOTO 110
100 INPUT 'ВВЕДИТЕ F0-1,F-10,F00,F10,F01,F11 'A,B,C,D,E,I
110 INPUT 'ВВЕДИТЕ ЗНАЧЕНИЯ X0,Y0 'X0,Y0
120 INPUT 'ВВЕДИТЕ X,Y 'X,Y
130 LET P=(X-X0)/H:LET Q=(Y-Y0)/L:IF N=4 THEN 160
140 IF N=6 THEN 170
150 LET F=(1-P-Q)*A+P*B+Q*C:GOTO 190
160 LET F=(1-P)*(1-Q)*A+P*(1-Q)*B+Q*(1-P)*C+P*Q*D:GOTO 190
170 LET F=Q*(Q-1)*A/2+P*(P-1)*B/2+(1+P*Q-P*P-Q*Q)*C
180 LET F=F+P*(P-2*Q+1)*D/2+Q*(Q-2*P+1)*E/2+P*Q*I
190 PRINT 'F(X,Y)='F:GOTO 120:END

```

Контрольные примеры. Интерполяция функции последствия

$$f(x, y) = \int_{-\infty}^{\infty} y^{-y^2 z^2} - z - x e^{-z} dz,$$

заданной таблично.

1. Интерполяция по трем точкам. Задано: $n=3$; $h=0,3$; $l=0,1$; $x_0=0,4$; $y_0=0$; $f_{00}=2,5$; $f_{01}=2,456$ и $f_{10}=1,429$. Получим $f(0,7; 0,05) = 1,407$; $f(0,4; 0,05) = 2,478$ и т. д.

2. Интерполяция по четырем точкам. Задано: $n=4$, $h=0,3$; $l=0,1$; $x_0=0,4$; $y_0=0$; $f_{00}=2,5$; $f_{10}=1,429$; $f_{01}=2,456$ и $f_{11}=1,4$. Получим $f(0,7; 0,05) = 1,4145$; $f(0,5; 0,08) = 2,1118$ и т. д.

3. Интерполяция по шести точкам. Задано: $n=6$, $h=0,3$; $l=0,1$; $x_0=0,7$; $y_0=0,05$; $f_{0-1}=1,429$; $f_{-10}=2,487$; $f_{00}=1,419$, $f_{10} = 0,995$; $f_{01}=1,4$; $f_{11}=0,981$. Получим $f(0,4; 0) = 2,502$; $f(0,5; 0,03) = 2,065857778$.

Многоинтервальная квадратичная интерполяция — аппроксимация функции двух переменных может использоваться, когда необходимо интерполировать или аппроксимировать таблично или графически заданные функции двух переменных. Такая интерполяция — аппроксимация удобна, например, для вычисления токов по заданным напряжениям у приборов с графически определенным семейством вольт-амперных характеристик (биполярных и полевых транзисто-

ров, тиристоров и т. д.). Алгоритм интерполяции — аппроксимации следующий.

1. Функция $f(x, y)$ задается в виде матрицы $F(I, J)$, где I — номер строки таблицы (или кривой графика), J — номер столбца (или точки на кривой). При нумерации I и J с 0 имеем $0 \leq I \leq N-1$ и $0 \leq J \leq M-1$, где N — число строк (кривых), M — число столбцов (точек на каждой кривой). Кроме того, задаются приращения (шаги) $\Delta x = h = \text{const}$ и $\Delta y = l = \text{const}$ и начальные значения $x_0 = a$ и $y_0 = b$.

2. Для каждого значения x и y вычисляются:

$$J = \text{int}((x-a)/h), \text{ если } J=0, \text{ то берем } J=1, \\ I = \text{int}((y-b)/l), \text{ если } I=0, \text{ то берем } I=1, \\ P = (x-a-h)/h, Q = (y-b-l)/l.$$

3. Для каждого $I=I-1, I, I+1$ (в правой части I соответствует вычисленному в п. 2) вычисляется значение

$$Z = P(P-1)F(I, J-1)/2 + (1-P^2)F(I, J) + P(P+1)F(I, J+1)/2.$$

Эти значения присваиваются переменным C, D и E . Таким образом, обеспечивается тройная квадратичная интерполяция по переменной x .

4. Выполняется квадратичная интерполяция по переменной y :

$$Z = Q(Q-1)C/2 + (1-Q^2)D + Q(Q+1)E/2.$$

Считаем $F(x, y) = Z$.

При аппроксимации неоднократно используются п. 2, 3 и 4 алгоритма. Общее время вычислений примерно в 4 раза больше, чем при квадратичной интерполяции — аппроксимация функций одной переменной.

Программа 4.10.

```

10 PRINT 'МНОГОИНТЕРВАЛЬНАЯ ДВУХМЕРНАЯ КВАДРАТИЧНАЯ'
20 PRINT 'ИНТЕРПОЛЯЦИЯ-АППРОКСИМАЦИЯ ФУНКЦИИ F(X,Y)'
30 INPUT 'ЗАДАЙТЕ ЧИСЛО ТОЧЕК КАЖДОЙ КРИВОЙ M=':M:LET M=M-1
40 INPUT 'ЗАДАЙТЕ ЧИСЛО КРИВЫХ N=':N:LET N=N-1: DIM F(N,M)
50 INPUT 'ЗАДАЙТЕ X0,Y0 'A,B
60 INPUT 'ЗАДАЙТЕ ПРИБЛИЖЕНИЯ ПО X И Y - H,L 'H,L
70 FOR I=0 TO N:FOR J=0 TO M
80 PRINT!3.0!'ВВЕДИТЕ Y,I',J:INPUT F(I,J)
90 NEXT J:NEXT I
100 INPUT 'ВВЕДИТЕ Y,X 'Y,X:GOSUB 4000
110 PRINT!F1.9!'F(X,Y)='Z:GOTO 100
4000 REM'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(X,Y)
4010 LET J=INT((X-A)/H):IF J=0 THEN LET J=1
4020 LET I=INT((Y-B)/L):IF I=0 THEN LET I=1
4030 LET P=(X-A-J*H)/H:LET Q=(Y-B-I*L)/L
4040 LET I=I-1:GOSUB 4080:LETC=Z
4050 LET I=I+2:GOSUB 4080:LETE=Z
4060 LET I=I-1:GOSUB 4080:LETD=Z
4070 LET Z=Q*(Q-1)*C/2+(1-Q*Q)*D+Q*(Q+1)*E/2:RETURN
4080 LET Z=P*(P-1)*F(I,J-1)/2+(1-P*P)*F(I,J)
4090 LET Z=Z+P*(P+1)*F(I,J+1)/2:RETURN:END

```

Пример. Пусть требуется провести интерполяцию для семейства вольт-амперных характеристик мощного полевого транзистора $(F(x, y) = I_c(U_c, U_g))$, где I_c — ток стока, $x = U_c$ — напряжение на стоке и $y = U_g$ — напряжение на затворе, если она задана таблицей ($N = M = 6$, $h = 10$, $l = 1$, $y_0 = 3$, $x_0 = 0$):

Метод простых итераций основан на представлении (4.4) в виде

$$x = f(x) \quad (4.5)$$

и многократным применением итерационной формулы $x_{n+1} = f(x_n)$ до тех пор, пока соблюдается условие

$$|x_{n+1} - x_n| \geq \epsilon, \quad (4.6)$$

Напряжение на затворе $y, В$	Напряжение на стоке $x, В$					
	0	10	20	30	40	50
3	0	0	0	0	0	0
4	0	0,35	0,4	0,45	0,5	0,55
5	0	0,8	1,1	1,2	1,22	1,25
6	0	1,25	1,8	2	2,12	2,15
7	0	1,8	2,7	3,15	3,17	3,2
8	0	2,95	4	4,45	4,5	4,7

Значения токов приведены в амперах. Введя эти данные, будем получать $F(35; 5,5) = 1,631875 А$; $F(45; 7,5) = 3,8375 А$ и т. д. Отметим, что этот метод обеспечивает высокую точность аппроксимаций для семейств достаточно сложных кривых, в том числе не монотонных.

где ϵ — заданная погрешность вычисления корня x . Итерационный процесс сходится (т. е. $x_n \rightarrow \bar{x}$ при $n \rightarrow \infty$), если соблюдается условие $f'(x) < 1$ при $a < x < b$.

Программа 4.11.

В строке 70 программы 4.11 записано выражение $f(x) = \sin x + 0,25$ (подпрограмма), соответствующее решению трансцендентного уравнения

$$F(x) = x - \sin x - 0,25 = 0. \quad (4.7)$$

Для начального значения $x = x_0 = 1,2$ и погрешности $\epsilon = E = 1 \cdot 10^{-6}$ получим $x = 1,71230493$ при времени счета около 5 с.

Метод Ньютона (касательных) основан на замене $F(x)$ в точке начального приближения $x = x_0$ касательной, пересечение которой с осью x дает первое приближение x_1 , и т. д. (см. рис. 4.4). Таким образом, итерационный процесс схождения к корню реали-

§ 4.3. Решение нелинейных и трансцендентных уравнений

Решение нелинейных (в частности, трансцендентных) уравнений вида

$$F(x) = 0 \quad (4.4)$$

заключается в отыскании одного или всех корней на отрезке $[a, b]$ изменения x . Обычно стараются локализовать каждый корень в своем отрезке $[a, b]$. Тогда нахождение

Программа 4.11.

```

10 PRINT 'РЕШЕНИЕ УРАВНЕНИЯ X=F(X) МЕТОДОМ'
15 PRINT 'ПРОСТЫХ ИТЕРАЦИЙ'
20 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X X0='X
25 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E='E
30 GOSUB 70
40 IF ABS(F-X)<E THEN 60
50 LET X=F:GOTO 30
60 PRINT 'КОРЕНЬ X='X: GOTO 20
65 REM 'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(X)'
70 LET F=SIN(X)+.25
75 RETURN:END
    
```

зуется формулой

$$x_{n+1} = x_n - F(x_n)/F'(x_n), \quad (4.8)$$

до тех пор, пока соблюдается условие (4.6). Метод обеспечивает быструю (квадратичную) сходимость, если

$$F(x_0) F''(x_0) > 0. \quad (4.9)$$

В качестве x_0 выбирают тот конец отрезка $[a, b]$, на котором знаки $F(x_0)$ и $F''(x_0)$ совпадают. Выигрыш во времени вычислений за счет быстрой сходимости уменьшается из-за необходимости вычисления помимо $F(x_n)$ производной $F'(x_n)$. Исключение составляют частные случаи, когда выражение, по которому вычисляется $F(x_n)/F'(x_n)$, не сложнее выражения для вычисления $F(x_n)$ отдельно.

Программа 4.12.

```

10 PRINT 'РЕШЕНИЕ УРАВНЕНИЯ F(X)=0 МЕТОДОМ НЬЮТОНА'
20 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ ПРИБЛИЖЕНИЕ X0='X
25 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E='E
30 GOSUB 60: LET X=X-F
40 IF ABS(F)>E THEN 30
50 PRINT 'КОРЕНЬ УРАВНЕНИЯ X='X:GOTO 20
55 REM 'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(X)/(DF/DX)'
60 LET F=(X-SIN(X)-.25)/(1-COS(X))
70 RETURN:END
    
```

Пример. Вычислить корень уравнения (4.7) на отрезке $[1, 1; 1, 2]$. В данном случае $F'(x) = 1 - \cos x$, так что

$$\frac{F(x_n)}{F'(x_n)} = \frac{x - \sin x - 0,25}{1 - \cos x}$$

Вычисление этого выражения оформлено подпрограммой, записанной со строки 60. Поскольку $F''(x) = \sin x > 0$ и $F(x) > 0$ при $x = b$, в качестве начального приближения возьмем $x_0 = b = 1, 2$. Тогда при погрешности $\varepsilon = 1 \cdot 10^{-6}$ получим $\bar{x} = 1, 171229656$ при времени счета $t_c \approx 3,5$ с.

Программа 4.13.

```

10 PRINT 'РЕШЕНИЕ УРАВНЕНИЯ F(X)=0 МОДИФИЦИРОВАННЫМ'
15 PRINT 'МЕТОДОМ НЬЮТОНА'
20 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ ПРИБЛИЖЕНИЕ X0='X
25 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E='E
30 GOSUB 70: LET L=F: LET X=X+E
40 GOSUB 70: LET L=E*L/(F-L): LET X=X-L-E
50 IF ABS(L)>E THEN 30
60 PRINT 'КОРЕНЬ УРАВНЕНИЯ X='X:GOTO 20
70 LET F=X-SIN(X)-.25
80 RETURN:END
    
```

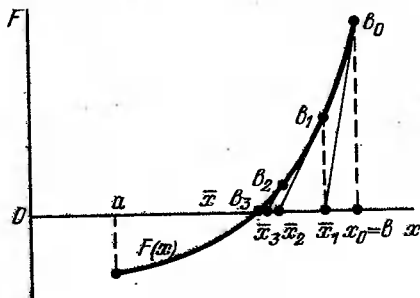


Рис. 4.4. Решение уравнения $F(x) = 0$ методом Ньютона (касательных)

Модифицированный метод Ньютона заключается в том, что вместо вычисления производной $F'(x_n)$ на каждом шаге итераций находится ее приближенное значение $F'(x_n) = dF(x_n)/dx \approx (F(x_n + \Delta x) - F(x_n))/\Delta x = \Delta F(x_n)/\Delta x$, где $\Delta x = \varepsilon$. Следовательно, итерационная формула имеет вид

$$x_{n+1} = x_n - \frac{\Delta x F(x_n)}{F(x_n + \Delta x) - F(x_n)}$$

Значение Δx не обязательно должно быть равно ε . Равенство $\Delta x = \varepsilon$ позволяет уменьшить число исходных данных при вводе.

Пример. Вычисление $F(x_n)$ выполняется подпрограммой, записанной со строки 70. Для приведенного выше примера расчет дает $x=1,171229653$ при $t_c \approx 4$ с. При модифицированном методе Ньютона отпадает необходимость вычисления $F'(x_n)$, но добавляется вычисление $F(x_n + \Delta x)$.

Метод Рыбакова также можно рассматривать как модификацию метода Ньютона при замене $F'(x_n)$ некоторым числом $M \geq F'(\xi)$, где ξ — значение x на отрезке $[a, b]$, при котором $F'(x)$ максимална. При $M \gg F'(\xi)$ сходимость не нарушается, но замедляется. Метод Рыбакова удобен для поиска всех корней уравнения (4.4) на отрезке $[a, b]$ с помощью следующего алгоритма [24].

1. Задаем начальные значения $x = x_0 = a$.
2. Для каждой n -й итерации ($n=0, 1, 2, \dots$) вычисляем

$$x_{n+1} = x_n + \frac{|F(x_n)|}{M} \quad (4.10)$$

и проверяем условие

$$x_{n+1} < b. \quad (4.11)$$

Если (4.11) не выполняется, значит, найдены все корни. В противном случае проверяем выполнение условия $|x_{n+1} - x_n| > \varepsilon$. Если оно выполняется, повторяем цикл вычислений по формуле (4.10). Если это условие не выполнено, значит, x_{n+1} есть один из корней и значение x_{n+1} выводится на печать. После этого переходим к выполнению следующего пункта.

3. Задаем начальное приближение к очередному корню $x_0 = x_{n+1} + \varepsilon$ и, если $x_0 < b$, идем к выполнению п. 2. Если $x_0 \geq b$, вычисления считаем законченными.

Число итераций при реализации метода Рыбакова $N = (b-a) M/\varepsilon$. Функция $F(x)$ на отрезке $[a, b]$ может иметь производную с разрывами первого рода.

Программа 4.14.

```

10 PRINT 'НАХОЖДЕНИЕ ВСЕХ КОРНЕЙ УРАВНЕНИЯ F(X)=0'
15 PRINT 'В ИНТЕРВАЛЕ ОТ A ДО B МЕТОДОМ РЫБАКОВА'
20 INPUT 'ВЕДТЕ ГРАНИЦЫ ИНТЕРВАЛА A, B 'A, B
25 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ НАХОЖДЕНИЯ КОРНЯ E='E
30 INPUT 'ЗАДАЙТЕ ЧИСЛО M='M: LET X=A: LET T=0: LET I=0
40 GOSUB 120: LET Z=X+ABS(F)/M
50 IF Z>=B GOTO 115
60 IF Z<=E GOTO 90
70 IF T=0 THEN LET W=Z
80 LET X=Z+E: LET Y=Z: LET T=1: GOTO 40
90 IF T<>1 THEN 110
100 LET I=I+1: PRINT 12.0: 'X' I '=' I 1.9: (Y+W)/2: LET T=0
110 LET X=Z: GOTO 40
115 PRINT 'КОНЕЦ ВЫЧИСЛЕНИЙ': STOP
120 LET F=X^4-13*X^2+36
130 RETURN: END

```

Пример. Найти все корни уравнения $F(x) = x^4 - 13x^2 + 36 = 0$ на отрезке $[-4, 4]$, т. е. $a = -4$, $b = 4$, с точностью $\varepsilon = 1 \cdot 10^{-3}$. Задав $M = 100$, получим: $\bar{x}_1 = -2,9987453$; $\bar{x}_2 = -1,999170328$; $\bar{x}_3 = 2,000132432$ и $\bar{x}_4 = 3,000785311$. Общее время вычислений около 2 мин.

Метод деления отрезка $[a, b]$ пополам (метод дихотомии) реализуется следующим алгоритмом (для $F(a) > 0$).

1. Находим $x = (a+b)/2$.
2. Вычисляем $F(x)$.
3. Если $F(x) > 0$, задаем $a = x$, иначе $b = x$.
4. Проверяем условие $b - a > \varepsilon$; если оно выполняется, идем к п. 1, если не выполняется, заканчиваем вычисления и считаем, что $x = x$ с заданной точностью ε .

Число итераций при использовании этого метода

$$N \approx \frac{\ln((b-a)/\varepsilon)}{\ln 2}$$

значительно, и поэтому сходимость его медленная. Однако при любой ширине отрезка $[a, b]$ сходимость гарантирована. Кроме того, простота реализации метода уменьшает число вспомогательных операций и частично компенсирует увеличение общего времени счета из-за медленной сходимости.

Программа 4.15.

Пример. Для уравнения $F(x) = E - x - R I_0 (\exp(Dx) - 1)$ при $E = 2$, $R = 10$, $I_0 = 1 \cdot 10^{-8}$, $D = 20$, $A = 0$, $B = 2$ и $\varepsilon = H = 1 \cdot 10^{-4}$. Получим $x = 0,8143920898$. При $F(a) < 0$ берем $F(x) := -F(x)$.

Метод поразрядного приближения для поиска всех корней отрезка $[a, b]$ реализуется следующим алгоритмом.

1. Задаем шаг $c = h$, $x = a$, $k = 0$ и находим $W = \text{sgn } F(x)$.
2. Задаем значение $x = (x+c)$ и проверяем условие $(x-c) \geq b$. Если оно выполняется, заканчиваем счет, иначе идем к п. 3.
3. Вычисляем $F(x)$ и проверяем условие $FW/c > 0$. Если оно выполняется, идем к п. 2, иначе к п. 4.
4. Задаем $c = -c/R$, где R — показатель разрядности (уменьшения шага c), и проверяем выполнение условия $|c| > \varepsilon/R$, где $\varepsilon =$

заданная погрешность вычисления корня. Если это условие выполняется, идем к п. 2, иначе к п. 5.

5. Задаем $k = k + 1$ и выводим на печать (индикацию) значение k -го корня $x_k = x$. Затем полагаем $c = h$, $W := -W$ и идем к п. 2.

Программа 4.15.

```

01 PRINT 'РЕШЕНИЕ УРАВНЕНИЕ F(X)=0 МЕТОДОМ ДИХОТОМИИ'
05 INPUT 'E, R, I0, D' E, R, I0, D
10 INPUT 'A, B, H' A, B, H
20 LET X=(A+B)/2
30 GOSUB 90
40 IF F>0 THEN 60
50 LET B=X: GOTO 70
60 LET A=X
70 IF B-A>H THEN 20
80 PRINT 'КОРЕНЬ=X: STOP
90 LET F=E-X-R*I0*(EXP(D*X)-1)
100 RETURN: END
    
```

Программа 4.16.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ВСЕХ КОРНЕЙ НЕЛИНЕЙНОГО УРАВНЕНИЯ F(X)=0'
20 PRINT 'В ИНТЕРВАЛЕ (A,B) МЕТОДОМ ПОРАЗРЯДНОГО ПРИБЛИЖЕНИЯ'
30 INPUT 'ЗАДАЙТЕ ГРАНИЦЫ ИНТЕРВАЛА A, B' A, B
40 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕНИЯ КОРНЕЙ E=' E
50 INPUT 'ЗАДАЙТЕ ШАГ НАЧАЛЬНОГО ПОИСКА H=' H
60 LET C=H: LET K=0: LET X=A: GOSUB 120: LET W=SGN(F)
70 LET X=X+C: IF X-C>B THEN PRINT 'КОНЕЦ': STOP
80 GOSUB 120: IF F*W/C>0 THEN 70
90 LET C=C/4: IF ABS(C)>E/4 THEN 70
100 LET K=K+1: PRINT I: 2.0! 'X' K! F! .9!' = X
110 LET C=H: LET W=-W: GOTO 70
120 LET I=.2718**X**EXP(-10**X)+1E-8*(EXP(20**X)-1)
130 LET F=(1-X)/125-I: RETURN: END
    
```

Пр и м е р. Найти напряжения на туннельном диоде из решения уравнения

$$F(U) = (E - U)/R - I(U) = 0, \quad (4.12)$$

где E — напряжение источника питания, R — сопротивление в его цепи и

$$I(U) = I = A U e^{-\alpha U} + D(e^{\beta U} - 1)$$

— N -образная вольт-амперная характеристика туннельного диода. Для $x = U$, $A = 0,2718$, $\alpha = 10$, $D = 1 \cdot 10^{-8}$, $\beta = 20$, $E = 1$ В и $R = 125$ Ом получаем подпрограмму, записанную в строках 120 и 130. Задав $a = 0$, $b = E$ и $\epsilon = E1 = 0,001$, получим $x_1 = 0,043$ В, $x_2 = 0,234$ В и $x_3 = 0,625$ В. Следовательно, в данном случае линия нагрузки резистора R пересекает вольт-амперную характеристику в трех точках.

Метод десятичного приближения аналогичен методу поразрядного приближения при $R = 10$. Метод дает все верные цифры результата в пределах заданной погрешности ϵ (остальные цифры — нули).

Метод хорд (см. рис. 4.5, а). При этом методе каждое значение x_{n+1} находится как точка пересечения оси абсцисс с хордой, проведенной через точки $F(a)$ и $F(b)$, причем одна из этих точек фиксируется — та, для которой знаки $F(x)$ и $F''(x)$ одинаковы. Если неподвижен конец хорды $x = a$, то

$$x_{n+1} = x_n - \frac{F(x_n)}{F(x_n) - F(a)} (x_n - a),$$

а если неподвижен конец хорды $x = b$, то

$$x_{n+1} = x_n - \frac{F(x_n)}{F(b) - F(x_n)} (b - x_n).$$

Если

$$|x_{n+1} - x_n| > \epsilon,$$

то в первом случае считаем $b = x_{n+1}$, во втором $a = x_{n+1}$ и повторяем вычисления. При использовании метода хорд полагаем, что корень \bar{x} находится на отрезке $[a, b]$.

Метод секущих (см. рис. 4.5, б) реализуется алгоритмом, описанным выше, если абсциссы a и b взяты с одной стороны от корня и не фиксируются.

Необходимость вычисления $F'(x)$ (условия сходимости этих методов аналогичны (4.9)) и выбора одной из двух формул затрудняют практическое применение методов хорд и секущих в отдельности.

Комбинированный метод секущих — хорд обеспечивает гарантированную сходимость при выборе в пределах отрезка $[a, b]$ двух приближений: нулевого x_0 и первого x_1 . Он реализуется алгоритмом, описанным для метода Ньютона с заменой производной $F'(x)$ ее приближенным значением — множителем перед $F(x_n)$:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{F(x_n) - F(x_{n-1})} F(x_n).$$

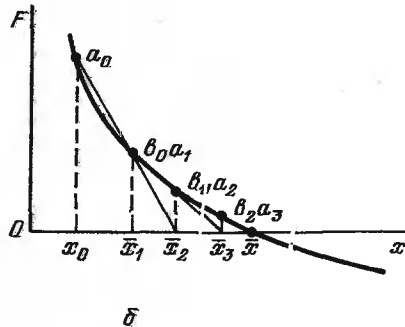
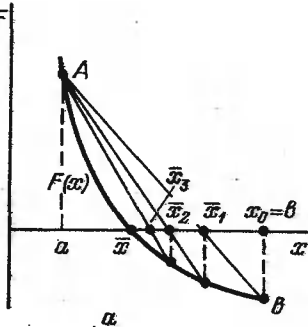


Рис. 4.5. Решение уравнения $F(x) = 0$ методом хорд (а) и секущих (б)

Программа 4.17.

```

01 PRINT 'РЕШЕНИЕ УРАВНЕНИЯ F(X)=0 КОМБИНИРОВАННЫМ'
02 PRINT ' МЕТОДОМ СЕКУЩИХ-ХОРД'
05 INPUT 'E, R, I0, D', E, R, I0, I,
10 INPUT 'X0, X1, H' X0, X1, H
20 LET X=X0: GOSUB 80
30 LET A=F: LET X=X1: GOSUB 80
40 LET B=F: LET Y=X0-A*(X1-X0)/(B-A)
50 LET X0=X1: LET X1=Y
60 IF ABS(X1-X0)>H THEN 20
70 PRINT 'КОРЕНЬ=X1: STOP
80 LET F=E-X-R*I0*(EXP(D*X))-1
90 RETURN: END
100 RETURN: END

```

Пример. Для уравнения $F(x) = x^3 + x^2 - x - 1$ при $x_0 = 2$, $x_1 = 1.5$ и $\varepsilon = 1 \cdot 10^{-5}$ находим $\bar{x} = 1$ при $t_c \approx 10$ с. Аналогичный результат получим при $x_0 = 0.5$ и $x_1 = 1.5$. Вычисление $F(x)$ оформляется подпрограммой, записанной со строки 90.

Метод Эйткена — Стеффенсона с ускоренной сходимость обеспечивает решение уравнения (4.5) по следующему алгоритму.

1. Задаем начальное приближение $x_n = x_0$.
2. Находим первое $x_1 = f(x_0)$ и второе $x_2 = f(x_1)$ приближения.
3. Вычисляем $x_{n+1} = (x_0 x_2 - x_1^2) / (x_0 - 2x_1 + x_2)$.

4. Проверяем условия $|x_{n+1} - x_n| > \varepsilon$, $x_0 - 2x_1 + x_2 \neq 0$. Если эти условия соблюдаются, идем к п. 1, т. е. задаем x_n новое значение x_{n+1} , в противном случае останавливаем счет и получаем $x = x_{n+1}$.

Метод Эйткена — Стеффенсона при сложных $f(x)$ имеет ускоренную сходимость (по сравнению с методом простых итераций). Однако при простых функциях $f(x)$ время счета практически не уменьшается, так как число дополнительных операций в этом методе существенно больше, чем в методе простых итераций.

Программа 4.18.

```

05 PRINT 'РЕШЕНИЕ УРАВНЕНИЯ X=F(X) МЕТОДОМ'
10 PRINT ' ЭЙТКЕНА-СТЕФФЕНСОНА'
15 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ ПРИБЛИЖЕНИЕ X0=X'
20 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E=' E
30 LET A=X: GOSUB 90: LET B=F: LET X=F
40 GOSUB 90: LET C=(A-2*B+F)
50 IF C=0 THEN 70
60 LET X=(A*F-B*B)/C
70 IF ABS(X-F)>E THEN 30
80 PRINT 'КОРЕНЬ УРАВНЕНИЯ X=' X: GOTO 20
90 LET F=SIN(X)+.25
100 RETURN: END

```

Пример. Используя данные к программе 4.11, при $\varepsilon = 1 \cdot 10^{-6}$ получим корень (4.7) $\bar{x} = 1,171229421$ при $t_c \approx 7$ с. Вычисление функции $f(x)$ выполняется подпрограммой, записанной в строках 90 и 100.

Метод обратной интерполяции — экстраполяции заключается в вычислении ряда значений $y_i = F(x_i)$ для заданных x_i на отрезке $[a, b]$. Затем, полагая $y = 0$, с помощью обратной интерполяции находим x_i (рис. 4.6, а). Корень может быть найден и за пределами отрезка $[a, b]$ (рис. 4.6, б). В последнем случае применяется обратная экстраполяция. Для реализации этого метода могут использоваться описанные в § 4.2 программы.

К сожалению, для произвольных $F(x)$ оценка погрешности этого метода отсутствует. Однако для получения результата с заданной погрешностью можно построить итерационную процедуру уточнения корня. Отметим, что при $m = 2$ отсчетах $F(x)$ данный метод фактически является комбинированным методом секущих — хорд, реализующим линейную интерполяцию — экстраполяцию.

Метод обратной квадратичной интерполяции — экстраполяции заключается в замене $F(x)$ полиномом Лагранжа второй степени (число отсчетов $m = 3$). При этом можно

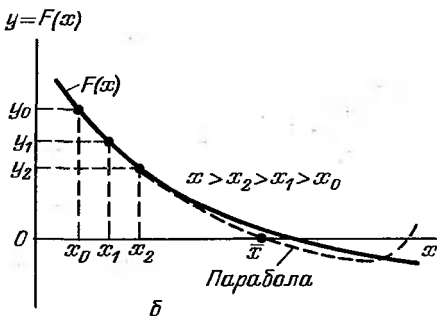
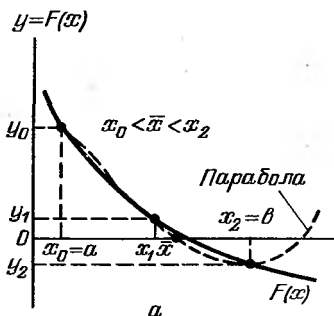


Рис. 4.6. Решение уравнения $F(x) = 0$ методом обратной интерполяции (а) и экстраполяции (б)

получить аналитическое выражение для приближенного значения корня.

Действительно, заменив x на y и y на x , полином Лагранжа второй степени можно представить в виде $x(y) = b_0 + b_1(y - y_0) +$

Сходимость данного алгоритма основана на свойстве интерполяционного полинома давать точные значения x при заданном y в узле интерполяции (т. е. в точке x_1).

Программа 4.19.

```

10 PRINT 'РЕШЕНИЕ УРАВНЕНИЯ F(X)=0 МЕТОДОМ КВАДРАТИЧНОЙ'
20 PRINT '      ИНТЕРПОЛЯЦИИ-ЭКСТРАПОЛЯЦИИ'
30 INPUT 'ЗАДАЙТЕ ГРАНИЦЫ А,В ' А,В
35 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕНИЙ E=' E
40 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ ПРИБЛИЖЕНИЕ X1=' C
50 LET X=A:GOSUB 120:LET L=F
60 LET X=B:GOSUB 120:LET M=F
70 LET X=C:GOSUB 120:LET Q=C
80 LET N=(C-A)/(F-L):LET P=(B-A)/(M-L)
90 LET C=A-L*(N+(N-P)*F/(M-F))
100 IF ABS(Q-C)>E THEN 70
110 PRINT 'КОРЕНЬ X=' C:STOP
120 LET F=X-SIN(X)-.25:RETURN:END

```

$+ b_2(y - y_0)(y - y_1)$. Для $y = 0$ находим

$$\bar{x} = b_0 - b_1 y_0 + b_2 y_0 y_1. \quad (4.13)$$

В соответствии с методом Эйткена и с учетом взаимной замены переменных x и y имеем

$$b_0 = x_0, \quad b_1 = \frac{x_1 - x_0}{y_1 - y_0},$$

$$b_2 = \frac{x_2 - x_1}{y_2 - y_0} - b_1 \frac{y_2 - y_1}{y_2 - y_0}.$$

Подставив b_0 , b_1 и b_2 в (4.13), после элементарных преобразований получим

$$\bar{x} = x_0 - y_0 \left(\frac{x_1 - x_0}{y_1 - y_0} + \frac{(x_1 - x_0)/(y_1 - y_0) - (x_2 - x_0)/(y_2 - y_0)}{y_2 - y_1} y_1 \right). \quad (4.14)$$

Приведем алгоритм вычислений этим методом (один из возможных).

1. Полагаем $x_0 = a$ и вычисляем $F(x_0) = y_0$.
2. Полагаем $x_2 = b$ и вычисляем $F(x_2) = y_2$.
3. Задаем начальное приближение к корню $x = x_1$ и вычисляем $F(x_1) = y_1$.
4. По формуле (4.14) находим корень \bar{x} (см. рис. 4.6).
5. Проверим условие $|\bar{x} - x_1| > \epsilon$; если оно выполняется, задаем $x_1 = \bar{x}$ и идем к п. 3, если не выполняется, останавливаем счет и считаем x корнем.

Пример. Используя данные программы 4.11, при $\epsilon = 1 \cdot 10^{-6}$ получим $\bar{x} = 1,174229661$ при $t_c \approx 4$ с.

§ 4.4. Решение систем нелинейных уравнений

Решение систем нелинейных уравнений может выполняться описанными выше методами, применяемыми поочередно к каждому из уравнений системы с контролем погрешности схождения каждой переменной к корню с заданной погрешностью. Остановимся на описании двух модификаций метода итераций.

Решение системы нелинейных уравнений методом простых итераций заключается в реализации итерационного процесса по следующей формуле:

$$X_{i(n+1)} = f_i(X_{i(n)}), \quad (4.15)$$

применяемой после преобразования системы нелинейных уравнений общего вида

$$F_i(X_i) = 0 \quad (4.16)$$

к виду $X_i = f_i(X_i)$. Здесь i — номер переменной (1, 2, ..., N), n — номер итерации. Вычисления ведутся до тех пор, пока соблюдается условие $|X_{i(n+1)} - X_{i(n)}| > \epsilon$, где ϵ — заданная точность.

Метод Зейделя отличается от метода простых итераций тем, что уточненные

$X_{i(n+1)}$ сразу подставляются в последующие уравнения. Обычно (но не всегда) метод Зейделя дает лучшую сходимость.

Необходимость преобразования (4.16) и трудности в обеспечении быстрой сходимости ограничивают применение этих методов. Однако программная реализация их довольно проста (см. Приложение 5).

Метод Ньютона (или Ньютона — Рафсона) является наиболее распространенным методом решения системы уравнений (4.16). Он реализуется следующим алгоритмом.

1. Задаем абсолютную или относительную погрешность $\varepsilon = E$, число уравнений N , максимальное число итераций M и вектор начальных приближений X_{i0} (с компонентами $x_{10}, x_{20}, \dots, x_{N0}$).

2. Используя разложение $F_i(X_i)$ в ряд Тейлора, формируем матрицу Якоби $[\partial F_i / \partial X_i]$, необходимую для расчета приращений $F_i(X_i)$ при малом изменении переменных. Матрица Якоби в развернутом виде записывается следующим образом:

$$\begin{bmatrix} \partial F_1 / \partial x_1 & \partial F_1 / \partial x_2 & \dots & \partial F_1 / \partial x_N \\ \partial F_2 / \partial x_1 & \partial F_2 / \partial x_2 & \dots & \partial F_2 / \partial x_N \\ \dots & \dots & \dots & \dots \\ \partial F_N / \partial x_1 & \partial F_N / \partial x_2 & \dots & \partial F_N / \partial x_N \end{bmatrix}$$

Поскольку аналитическое дифференцирование $F_i(X_i)$ в общем случае нежелательно, заменяем частные производные в матрице Якоби их приближенными конечно-разностными значениями

$$\frac{\partial F_i}{\partial X_i} \approx \frac{F_i(X_i + H) - F_i(X_i)}{H_i}$$

где H_i — малое приращение X_i , например $H_i = \varepsilon |X_i|$.

3. Составляем и решаем систему линейных уравнений для малых приращений X_i :

$$\begin{bmatrix} \partial F_1 / \partial x_1 & \partial F_1 / \partial x_2 & \dots & \partial F_1 / \partial x_N \\ \partial F_2 / \partial x_1 & \partial F_2 / \partial x_2 & \dots & \partial F_2 / \partial x_N \\ \dots & \dots & \dots & \dots \\ \partial F_N / \partial x_1 & \partial F_N / \partial x_2 & \dots & \partial F_N / \partial x_N \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \dots \\ \Delta x_N \end{bmatrix} = \begin{bmatrix} -F_1 \\ -F_2 \\ \dots \\ -F_N \end{bmatrix}$$

Решение этой системы дает $\Delta x_1, \Delta x_2, \dots, \Delta x_N$, т. е. ΔX_i .

4. Вычисляем уточненные значения

$$\begin{aligned} x_{1(n+1)} &= x_{1(n)} + \Delta x_1, \\ x_{2(n+1)} &= x_{2(n)} + \Delta x_2, \\ \dots & \dots \dots \\ x_{N(n+1)} &= x_{N(n)} + \Delta x_N. \end{aligned}$$

или в общем виде

$$X_{i(n+1)} = X_{in} + \Delta X_i.$$

5. Для всех ΔX_i проверяем одно из условий: $|\Delta X_i| > \varepsilon$, $|\Delta X_i / X_i| > \varepsilon$.

Если оно выполняется, идем к п. 2, т. е. выполняем новую итерацию. Иначе считаем вектор $X_{i(n+1)}$ найденным решением.

Отметим, что решение системы нелинейных уравнений (4.16) можно представить и в виде $X_{i(n+1)} = X_{in} - W_{(n)}^{-1} F_i(X_{in})$, где $W_{(n)}^{-1}$ — обратная матрица Якоби. Обращение матрицы Якоби осуществляется в ходе решения системы линейных уравнений для приращений ΔX_i методом Гаусса.

Программа 4.20.

```

10 PRINT 'РЕШЕНИЕ СИСТЕМЫ НЕЛИНЕЙНЫХ УРАВНЕНИЙ'
20 PRINT ' МОДИФИЦИРОВАННЫМ МЕТОДОМ НЬЮТОНА'
30 INPUT 'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N=':N:DIM A(N,N),B(N),X(N),F(N)
40 INPUT 'ЗАДАЙТЕ МАКСИМАЛЬНОЕ ЧИСЛО ИТЕРАЦИЙ M=':M
50 INPUT 'ЗАДАЙТЕ ОТНОСИТЕЛЬНОМУ ПОГРЕШНОСТЬ E=':E:LETS=0
60 FOR I=1 TO N:PRINT!2.0!'ВВЕДИТЕ X'I'(0)!'
70 INPUT X(I):NEXT I
80 GOSUB 260:FOR I=1 TO N:LETB(I)=-F(I):NEXT I
90 FOR J=1 TO N:LETX(J)=X(I):LETH=E*ABS(X)
100 LETX(J)=X+H:GOSUB 260:FOR I=1 TO N
110 LETA(I,J)=(F(I)+B(I))/H:NEXT I:LETX(J)=X:NEXT J
120 LETS=S+1:IF S=M+1 THEN PRINT'ЧИСЛО ИТЕРАЦИЙ S=M':STOP
130 FOR I=1 TO N-1:FOR J=I+1 TO N
140 LETA(J,I)=-A(J,I)/A(I,I):FOR K=I+1 TO N
150 LETA(J,K)=A(J,K)+A(J,I)*A(I,K):NEXT K
160 LETB(J)=B(J)+A(J,I)*B(I):NEXT J:NEXT I
170 LETF(N)=B(N)/A(N,N):FOR I=N-1 TO 1 STEP -1
180 LETH=B(I):FOR J=I+1 TO N:LETH=H-F(J)*A(I,J):NEXT J
190 LETF(I)=H/A(I,I):NEXT I:LETR=0
200 FOR I=1 TO N:LETX(I)=X(I)+F(I)
210 IF ABS(F(I)/X(I))>E THEN LETR=1
220 NEXT I:IF R=1 THEN 80
230 PRINT'РЕШЕНИЕ СИСТЕМЫ'
240 FOR I=1 TO N:PRINT!2.0!'X'I'!F1.9!'=X(I):NEXT I
245 PRINT!2.0!'ЧИСЛО ИТЕРАЦИЙ S=':S:STOP
250 REM'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(I)=F(X(1),X(2),...,X(N))'
260 LETF(1)=X(1)+3*MLT(X(1))-X(2)*X(2)
270 LETF(2)=2*X(1)*X(1)-X(1)*X(2)-5*X(1)+1
280 RETURN:END

```

$$x_{1,2} = -\frac{a_1}{2a_2} \pm \sqrt{\left(\frac{a_1}{2a_2}\right)^2 - \frac{a_0}{a_2}}$$

записывается подпрограмма вычисления частей (4.16) в виде

$F(I)$ =(Функция переменных $X(I)$).

В данном случае подпрограмма служит для решения системы уравнений

$$x_1 + 3 \lg x_1 - x_2^2 = 0,$$

$$2x_1 - x_1x_2 - 5x_1 + 1 = 0.$$

Задав $N=2$, $M=10$ (максимальное число итераций для случая, если решение расходуется), $\varepsilon = E = 1 \cdot 10^{-4}$, $x_{10} = X(0) = 3,4$ и $x_{20} = X(0) = 2,2$, получим $x_1 = 3,487442788$, $x_2 = 2,261628631$ и $S=3$. Значение переменной S дает число итераций, а значения $F(I)$ — невязки системы. Время счета контрольного примера 16 с.

Решение (4.16) с помощью методов минимизации функций ряда переменных описано в § 4.6.

§ 4.5. Решение алгебраических уравнений с действительными и комплексными коэффициентами

Алгебраические (но не трансцендентные) уравнения сводятся к канонической форме

$$A_0 Z^n + A_1 Z^{n-1} + \dots + A_{n-1} Z + A_n = 0, \quad (4.17)$$

где коэффициенты A_0, A_1, \dots, A_n — в общем случае комплексные числа. В таком виде алгебраические уравнения используются редко. На практике обычно встречаются полиномиальные уравнения с действительными коэффициентами

$$a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n = 0, \quad (4.18)$$

причем нередко (4.18) записывается в другой форме:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0. \quad (4.19)$$

Отметим некоторые известные свойства алгебраических уравнений, на которых базируются методы их решения.

1. Корни алгебраических уравнений могут быть действительными и комплексно сопряженными, причем на комплексной плоскости все они лежат в пределах кольца, ограниченного окружностями с радиусами $R = 1 + |a_{\max}/a_n|$ и $r = 1 + |a'_{\max}/a_n|$, где a_{\max} — наибольший по модулю коэффициент полинома, a'_{\max} — следующий наибольший по модулю коэффициент (т. е. без учета a_{\max}).

2. Число корней с учетом их кратности равно n , причем полиномы нечетной степени обязательно имеют хотя бы один действительный корень.

3. Число положительных (отрицательных) корней равно числу перемен (постоянств) знаков в последовательности a_n, a_{n-1}, \dots, a_0 коэффициентов или меньше его на четное число.

Квадратное уравнение

$$a_2 x^2 + a_1 x + a_0 = 0 \quad (4.20)$$

причем если подкоренное выражение отрицательно, корни получаются комплексно сопряженными, если оно положительно, они действительные.

Программа 4.21.

Пример. Для уравнения $2x^2 - 5x - 10 = 0$ получим $x_1 = -1,311737691$ и $x_2 = 3,811737691$, а для уравнения $x^2 + 2x + 15 = 0$ получим $x_{1,2} = -1 \pm j \cdot 3,741657387$.

Приведенное кубическое уравнение

$$x^3 + a_2 x^2 + a_1 x + a_0 = 0 \quad (4.21)$$

может решаться по точным формулам [18]. Однако более компактное решение получается при использовании следующего алгоритма.

Вначале методом половинного деления с $\varepsilon = 1 \cdot 10^{-10}$ находится действительный корень x_3 (он обязательно существует). Затем делением (4.21) на $(x - x_3)$ получается квадратное уравнение, из которого вычисляются два других корня x_1 и x_2 .

Программа 4.22.

Пример. Для уравнения $x^3 - 6x^2 + 21x - 15 = 0$ получаем $x_{1,2} = 1 \pm j \cdot 3,464101615$ и $x_3 = 4$, а для уравнения $x^3 - 6x^2 + 11x - 6 = 0$ получим $x_1 = 3$, $x_2 = 2$ и $x_3 = 1$.

Уравнение (4.19) с действительными коэффициентами решается методом Хичкока. При этом (вначале) итерационным методом из (4.19) выделяется квадратичный множитель (если $n \geq 2$) и вычисляются два его корня. Затем (4.19) делится на этот множитель, т. е. n уменьшается на 2, и процесс повторяется до вычисления всех корней.

Программа 4.23.

Пример. Для уравнения $x^4 + 9x^3 + 31x^2 + 59x + 60 = 0$ получим корни $x_1 = -1 + j \cdot 2$, $x_2 = -1 - j \cdot 2$, $x_3 = -3 + j \cdot 0$, $x_4 = -4 + j \cdot 0$ (т. е. x_3 и x_4 действительны), а для уравнения $x^5 + 8x^4 + 31x^3 + 80x^2 + 94x + 20 = 0$ получим $x_1 = -0,2679491924$, $x_2 = -2$, $x_3 = -1 + j \cdot 3$, $x_4 = -1 - j \cdot 3$ и $x_5 = -3,732050808$ (при вычислениях принималось $\varepsilon = 1 \cdot 10^{-6}$).

Решение уравнения общего вида (4.17) с комплексными коэффициентами $A_i = a_i + jb_i$ обобщенным методом Ньютона реализуется следующим алгоритмом.

1. Вводим массив A_i и задаем точность вычислений ε .

2. Задаем начальное приближение $Z_0 = x_0 + jy_0$, в частности $x_0 = 0$ и $y_0 = 0$ (метод дает сходимость к какому-либо корню при любых начальных приближениях).

3. По методу Ньютона (но с применением аппарата комплексной арифметики) находим корень (4.17) Z_1 .

4. Делим (4.17) на $(Z - Z_1)$, т. е. понижаем степень (4.17) на единицу и повторяем вычисления с п. 2 до нахождения второго корня и т. д. до тех пор, пока не будут вычислены все корни.

Более подробное описание этого метода см. в [24]. При использовании програм-

Программа 4.21.

```

05 PRINT 'РЕШЕНИЕ КВАДРАТНОГО УРАВНЕНИЯ'
10 INPUT 'ВВЕДИТЕ A2='C': INPUT 'ВВЕДИТЕ A1='B
15 INPUT 'ВВЕДИТЕ A0='A
20 LET D=B/2/C: LET F=D*D-A/C
30 IF F>0 THEN 60
40 PRINT 'X1,2='-D'+- J*'SQR(-F)
50 GOTO 10
60 PRINT 'X1='-D-SQR(F)
70 PRINT 'X2='-D+SQR(F)
80 GOTO 10:END

```

Программа 4.22.

```

05 PRINT 'РЕШЕНИЕ ПРИВЕДЕННОГО КУБИЧЕСКОГО УРАВНЕНИЯ'
10 INPUT 'ВВЕДИТЕ A2='C': INPUT 'ВВЕДИТЕ A1='B
15 INPUT 'ВВЕДИТЕ A0='A: INPUT 'ВВЕДИТЕ /A MAX/= 'K
20 LET X=0: LET K=K+1
25 LET G=X+C: LET L=G*X+B
30 IF L*X+A>=0 THEN 50
40 LET X=X+K
50 LET K=K/2: LET X=X-K
60 IF K>=1E-10 THEN 25
70 LET E=-G/2: LET M=E*L-E
80 IF M>=0 THEN 130
90 PRINT 'X1,2='E'+- J*'SQR(-M)
100 PRINT 'X3='X: STOP
130 LET F=SQR(M): LET H=F+E: LET I=E-F
140 PRINT 'X1='H: PRINT 'X2='I: PRINT 'X3='X:END

```

Программа 4.23.

```

05 PRINT 'ВЫЧИСЛЕНИЕ ВСЕХ КОРНЕЙ ПОЛИНОМА'
10 PRINT 'С ДЕЙСТВИТЕЛЬНЫМИ КОЭФФИЦИЕНТАМИ'
15 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ E='E
20 INPUT 'ЗАДАЙТЕ СТЕПЕНЬ ПОЛИНОМА N='N: DIM A(2*N+1)
25 FOR J=1 TO N+1: PRINT 'ВВЕДИТЕ A'12.0!N+1-J
30 INPUT A(J): NEXT J: LET K=1
35 LET T=1: LET C=A(2)/A(1)
40 IF N=1 THEN LET P=-C: LET Q=0: GOTO 270
50 IF N=2 THEN LET H=C*C/4-A(3)/A(1): GOTO 220
60 LET M=10: LET C=4: LET D=8: LET U=4
70 LET V=8: LET F=1: LET W=2: LET T=0
80 IF M>10 THEN 100
85 LET P=C: LET M=0: LET Q=D: LET C=U: LET D=V
90 LET U=P: LET V=Q: LET Y=C: LET Z=D: LET F=-F
100 LET M=M+1: LET H=0: LET Q=A(1): LET P=A(2)-C*Q: LET L=Q
120 FOR J=3 TO N: LET R=P: LET P=A(J)-C*R-D*Q
130 LET Q=R: LET R=L: LET L=Q-C*R-H*D: LET H=R: NEXT J
140 LET Q=A(N+1)-D*Q: LET S=L+C*R
150 IF T=0 THEN LET X=D/R: LET H=R*X+S*L: IF H=0 THEN 80
160 LET C=C+(P*S-Q*R)/H: LET D=D+(P*X+Q*L)/H
170 IF C-Y+D-Z<>0 THEN 190
180 IF F=-W THEN PRINT 'НЕТ РЕШЕНИЯ': STOP
185 LET W=-F
190 LET H=C*C/4-D: IF SQR((Q-P*C/2)^2+P*P*ABS(H))>E/N THEN 80
200 LET T=0: LET A(2)=A(2)-C*A(1): FOR J=3 TO N-1
210 LET A(J)=A(J)-C*A(J-1)-D*A(J-2): NEXT J
220 LET P=-C/2: LET Q=SQR(ABS(H))
230 IF H>=0 THEN LET M=P+Q: LET P=P-Q: LET Q=0: GOTO 250
240 LET M=P
250 PRINT!2.0!'X('K')='!F1.9!M' +J*('Q')': LET K=K+1
270 PRINT!2.0!'X('K')='!F1.9!P' +J*('-Q')': LET K=K+1
290 IF T<>0 THEN 350
300 LET N=N-2: LET A=0
310 IF SQR((S-R*C/2)^2+R*R*ABS(H))<=E THEN LET A=1
320 LET B=0: IF N>=2 THEN LET B=1
330 IF A+B=2 THEN LET T=1: GOTO 110
340 GOTO 35
350 PRINT 'ВЫЧИСЛЕНИЯ ЗАКОНЧЕНЫ':END

```

мы 4.22, реализуя этот метод, вводится: степень полинома $n=N$, погрешность равенства нулю производной $E1$, погрешность вычисления корней $\epsilon=E2$ и массив действительных a_i и мнимых b_i частей коэффициентов A_i . Если $A_i=a_i$ — действительные числа, задаем $b_i=0$.

Программа 4.24.

Пример. Решить уравнение

$$Z^6 + 8Z^4 + 31Z^3 + 80Z^2 + 94Z + 20 = 0.$$

Вводим $N=5$, $a_0=1$, $a_1=8$, $a_2=31$, $a_3=80$, $a_4=94$, $a_5=20$ и $b_0 \div b_5=0$. Результат получаем (при $E1=1 \cdot 10^{-3}$ и $E2=1 \cdot 10^{-5}$) с округлением на пятой цифре:

$$\begin{aligned} \operatorname{Re}Z(1) &= -2,67949 \cdot 10^{-1} \\ \operatorname{Re}Z(2) &= -2,00000 \\ \operatorname{Re}Z(3) &= -3,73205 \\ \operatorname{Re}Z(4) &= -1,00000 \\ \operatorname{Re}Z(5) &= -1,00000 \\ \operatorname{Im}Z(1) &= -7,96970 \cdot 10^{-17} \\ \operatorname{Im}Z(2) &= 1,51995 \cdot 10^{-16} \\ \operatorname{Im}Z(3) &= -4,61589 \cdot 10^{-17} \\ \operatorname{Im}Z(4) &= 3,00000 \\ \operatorname{Im}Z(5) &= -3,00000 \end{aligned}$$

$$\begin{aligned} K(1) &= 1 \\ K(2) &= 1 \\ K(3) &= 1 \\ K(4) &= 1 \\ K(5) &= 1 \end{aligned}$$

Значения $K(i)$ указывают на кратность i -го корня. Для $\operatorname{Im}Z(1)$, $\operatorname{Im}Z(2)$ и $\operatorname{Im}Z(3)$ значения меньше заданной погрешности $E2$, поэтому их следует считать нулевыми. Таким

Программа 4.24.

```

05 PRINT 'ВЫЧИСЛЕНИЕ КОРНЕЙ ПОЛИНОМА P(Z)'
10 PRINT 'С КОМПЛЕКСНЫМИ КОЭФФИЦИЕНТАМИ'
15 INPUT 'ЗАДАЙТЕ СТЕПЕНЬ ПОЛИНОМА N= ' N:LETH=N
20 DIM C(N+1),D(N+1),E(N+1),F(N+1),G1(N+1)
25 DIM A(N+1),B(N+1),F1(N+1),K(N),D1(N),Z1(N)
30 LET X0=0:LET Y0=0:INPUT 'ВВЕДИТЕ E1,E2 ' E1,E2
40 FOR I=1 TO N+1:PRINT!2.0!'ВВЕДИТЕ A('I-1'),B('I-1')'
50 INPUT A(I),B(I):NEXT I:LETH=1
120 LET X=X0:LET Y=Y0:LETS=1:FOR I=1 TO N+1
140 LET F(I)=A(I):LET G1(I)=B(I):LETC(I)=A(I):LETD(I)=B(I)
150 NEXT I:LETM=N:GOTO 600
170 LET U1=0:LET V1=0:LETL=1:GOTO 670
190 FOR I=1 TO N+L-1
200 LET F(I)=C(I):LET G1(I)=D(I):NEXT I
220 LET M=N-L:LETS=2:GOTO 600
230 IF SQR(U^2+V^2)>=E1 THEN 260
240 LET L=L+1:GOTO 670
260 LET Z=U^2+V^2
270 LET G=- (U1*U+V1*V)/Z
280 LET W=(U1*V-U*U1)/Z:LET Z=SQR(G*G+W*W):LET W1=W/Z
300 IF ABS(W/Z)>1 THEN LET W1=SIGN(W/Z)
310 LET F2=ASN(W1)/L
320 IF G<=0 THEN LET F2=#PI/L-ASN(W1)/L
330 LET Z=EXP(LOG(Z)/L):LET G=Z*COS(F2)
340 LET W=Z*SIN(F2):LETT=1:LETS=3:LETM=N
350 FOR I=1 TO N+1
360 LET F(I)=A(I):LET G1(I)=B(I):NEXT I
370 LET X=X0+T*G:LET Y=Y0+T*W:GOTO 600
390 IF (U^2+V^2)<(U1^2+V1^2) THEN 405
400 LET T=T/2:GOTO 370
410 IF T*SQR(G^2+W^2)<E2 THEN 460
420 LET X0=X:LET Y0=Y:LETS=1:FOR I=1 TO N+1
440 LET C(I)=A(I):LETD(I)=B(I):NEXT I:GOTO 170
460 LET K(I1)=L:LETD1(I1)=X:LET Z1(I1)=Y
470 FOR J=1 TO L:LETO=N-J+1:GOTO 700
500 FOR I=1 TO O:LETA(I)=E(I):LETB(I)=F1(I):NEXT I:NEXT J
520 IF N-L<=0 THEN 560
530 LET I1=I1+1:LETM=N-1:GOTO 120
560 FOR I=1 TO H
570 PRINT!2.0!'RE Z('I')='!F1.5!D1(I)!2.0!'IM Z('I')='!F1.5!Z1(I)
580 PRINT!2.0!'K('I')='K(I):NEXT I:STOP
600 LET U=F(1):LET V=G1(1)
610 FOR I=2 TO M+1
620 LET Z=U*X-U*Y+F(I):LET U=U*Y+U*X+G1(I):LET V=Z
640 NEXT I:IF S=1 GOTO 170
650 IF S=2 GOTO 230
660 GOTO 390
670 FOR I=1 TO M
680 LET C(I)=(M-I+1)*C(I):LETD(I)=(M-I+1)*D(I)
690 NEXT I:GOTO 190
700 LET E(1)=A(1):LET F1(1)=B(1)
710 FOR I=2 TO O
720 LET E(I)=A(I)+E(I-1)*X-F1(I-1)*Y
730 LET F1(I)=B(I)+E(I-1)*Y+F1(I-1)*X
740 NEXT I:GOTO 500:END

```

образом, уравнение имеет три действительных корня $Z_1 = -0,267949$, $Z_2 = -2$ и $Z_3 = -3,73205$ и два комплексно сопряженных $Z_{4,5} = -1 \pm j \cdot 3$. Время счета контрольного примера около 10 мин.

§ 4.6. Поиск экстремумов функций одной и множества переменных

На практике часто необходимо найти экстремум (или экстремумы) некоторой целевой функции $F(x_1, x_2, \dots, x_n)$ n переменных x_i (проектных параметров). Такая функция описывает $(n+1)$ -мерную поверхность. Соответственно функция $F(x)$ одного параметра $x_1 = x$ описывает некоторую кривую на плоскости (рис. 4.7, а). Поиск экстремумов функций одной переменной является самостоятельной и часто встречаемой задачей. Кроме того, к нему сводится гораздо более сложная задача поиска экстремумов функций множества переменных [3, 41].

В общем случае функция $F(x)$ может иметь несколько экстремумов (максимумов или минимумов). Из них главный (оптимальное решение для пространства проектирования) называется глобальным. Задача поиска экстремумов сводится к их локализации и уточнению значений x и $F(x)$ в точке экстремума. В дальнейшем для функций одной переменной под экстремумом будем подразумевать максимум $F(x)$. Поскольку максимуму функции $F(x)$ соответствует минимум функции $-F(x)$, то, сменив знак у $F(x)$, программами поиска максимума можно пользоваться и для поиска минимума функций. Будем также полагать, что на изменения x (если это особо не оговорено) накладываются ограничения в виде неравенств $a \leq x \leq b$, где a и b — границы интервала поиска. В пределах отрезка $[a, b]$ функцию считаем унимодальной, т. е. содержащей один максимум (рис. 4.7, б).

Метод равномерного поиска основан на том, что переменной x присваиваются значения $x + \Delta x$ с шагом $\Delta x = \text{const}$ и вычисляются значения $F(x)$. Если $F(x_{n+1}) > F(x_n)$, переменной x дается новое приращение. Как только $F(x_{n+1})$ станет меньше $F(x_n)$, поиск останавливается. При малой заданной погрешности этот метод неэкономичен по затратам машинного времени.

Метод порядного приближения является разновидностью метода равномерного

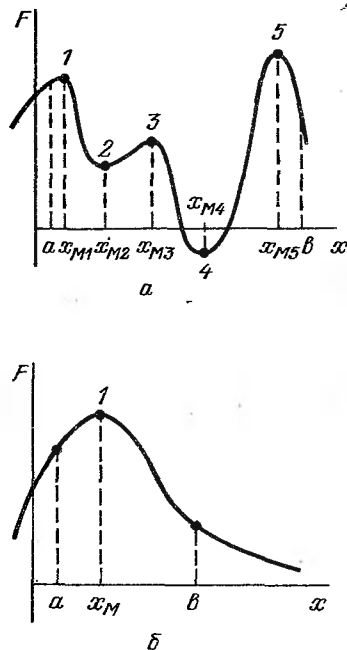


Рис. 4.7. Функция с несколькими экстремумами (а) и унимодальная функция с одним экстремумом максимумом (б)

поиска и реализуется следующим алгоритмом.

1. Задаем начальное приближение $x = x_0$ слева от максимума $F(x)$ и вычисляем $F(x_0)$. Задаем $D = h$, где $h = \Delta x$ — начальный шаг поиска.

2. Полагаем $G = F(x_n)$, где вначале $F(x_n) = F(x_0)$, задаем $x = x + D$ и вычисляем $F(x_{n+1}) = F(x)$.

3. Проверим условие $F(x_{n+1}) > G$; если оно выполняется, идем к п. 3, если нет — к п. 4.

4. Полагаем $D = -D/4$. Проверим условие $|D| > E/4$, где E — заданная погрешность вычисления x_m в точке максимума. Если оно выполняется, идем к п. 2, т. е. обеспечиваем поиск максимума в другом направлении с шагом в 4 раза меньше прежнего. Если данное условие выполняется, заканчиваем счет.

Программа 4.25.

```

10 PRINT 'ПОИСК МАКСИМУМА F(X) МЕТОДОМ'
15 PRINT 'ПОРЯДНОГО ПРИБЛИЖЕНИЯ'
20 INPUT 'ВВЕДИТЕ НАЧАЛЬНЫЙ ШАГ ПОИСКА H= ' H
25 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E= ' E
30 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X X0= ' X
35 LET D=H:GOSUB 80
40 LET G=F:LET X=X+D:GOSUB 80
50 IF F>G THEN 40
60 LET D=-D/4: IF ABS(D)>E/4 THEN 40
70 PRINT 'X_M=' X:PRINT 'F(X_M)= ' F:STOP
80 LET F=(.1*X-2)*X+10)*X
90 RETURN:END

```

Пример. Найти максимум функции

$$F(x) = 0,1x^3 - 2x^2 + 10x. \quad (4.22)$$

Подпрограмма вычисления $F(x)$ записана со строки 80. Задав $h=1$, $E=0,001$ и $x_0=2$, получим $x_m=3,33203125$ и $F(x_m)=14,81481312$.

Метод дихотомии (деления интервала поиска $[a, b]$ пополам) реализуется следующим алгоритмом.

1. Проверяем условие $|b-a| < 2E$, где E — заданная погрешность вычисления x_m . Если это условие выполняется, идем к п. 6; если не выполняется, идем к п. 2.

2. Делим интервал поиска $[a, b]$ пополам и вычисляем две абсциссы, симметрично расположенные относительно точки $x = (a+b)/2$: $x_1 = (a+b-e)/2$ и $x_2 = (a+b+e)/2$.

3. Для этих значений x вычисляем $F(x_1)$ и $F(x_2)$.

4. Проверяем условие $F(x_1) > F(x_2)$. Если оно выполняется, полагаем $b=x_2$ и идем к п. 1. Если не выполняется, идем к п. 5.

5. Полагаем $a=x_1$ и идем к п. 1.

6. Выводим на печать $x_m = (a+b)/2$ и вычисляем $F(x_m)$.

Программа 4.26.

```
10 PRINT'ПОИСК МАКСИМУМА F(X) МЕТОДОМ ДИХОТОМИИ'
20 INPUT'ВВЕДИТЕ ГРАНИЦЫ ИНТЕРВАЛА ПОИСКА A,B' A,B
25 INPUT'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E=' E
30 IF ABS(B-A)<2*E THEN 100
40 LETC=(A+B-E)/2:LETD=(A+B+E)/2
50 LETX=C:GOSUB 120:LETK=F
60 LETX=D:GOSUB 120:LETL=F
70 IF K>L THEN 90
80 LETA=C:GOTO 30
90 LETB=D:GOTO 30
100 LETX=(A+B)/2:GOSUB 120
110 PRINT'XM='X:PRINT'F(XM)='F:STOP
120 LETF=((.1*X-2)*X+10)*X
130 RETURN:END
```

Пример. Со строки 120 записана подпрограмма поиска максимума функции $F(x)$ по формуле (4.22). При $a=2$, $b=5$ и $E=0,001$ получим $x_m=3,333429565$ и $F(x_m)=14,81481481$.

Метод золотого сечения основан на делении отрезка $[a, b]$ по правилу золотого сечения

(см. алгоритм ниже). Это позволяет сразу отрезать $[a, b]$, каждый раз вычисляя лишь одно значение $F(x)$, а не два, как в методе дихотомии. Данный метод реализуется следующим алгоритмом.

1. Находим коэффициент дробления $k = (\sqrt{5}-1)/2$ отрезка $[a, b]$.

2. Находим абсциссу $x_1 = a + (1-k)(b-a)$ и вычисляем $F(x_1)$.

3. Находим абсциссу $x_2 = a + k(b-a)$ и вычисляем $F(x_2)$.

4. Проверяем выполнение условия $|x_2 - x_1| < E$, где E — заданная погрешность вычисления x_m . Если это условие выполняется, вычисляем $x_m = (x_1 + x_2)/2$ и $F(x_m)$, после чего останавливаем счет с выдачей значений x_m и $F(x_m)$. Если данное условие не выполняется, идем к п. 5.

5. Проверяем условие $F(x_1) < F(x_2)$. Если оно выполняется, полагаем $a=x_1$, $x_1=x_2$ и $F(x_1)=F(x_2)$, после чего выполняем п. 3 и п. 4.

6. Если $F(x_1) \geq F(x_2)$, полагаем $b=x_2$, $x_2=x_1$, $F(x_2)=F(x_1)$, после чего выполняем п. 2 и п. 4.

Программа 4.27.

Пример. Со строки 180 записана подпрограмма вычисления $F(x)$ для функции

(4.22). Для $a=2$, $b=5$ и $E=0,001$ получим $x_m=3,33495386$ и $F(x_m)=14,81481479$.

Метод квадратичной интерполяции — экстраполяции заключается в замене $F(x)$ в промежутке $x_1 \pm h$, где x_1 — начальное приближение, квадратической параболы, экстре-

Программа 4.27.

```
10 PRINT'ПОИСК МАКСИМУМА F(X) МЕТОДОМ ЗОЛОТОГО СЕЧЕНИЯ'
20 INPUT'ВВЕДИТЕ ПРЕДЕЛЫ ИЗМЕНЕНИЯ X A,B' A,B
25 INPUT'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E=' E
30 LETK=(SQRT(5)-1)/2
40 GOSUB 140
50 GOSUB 160
60 IF ABS(D-C)<E THEN 120
70 IF L>M THEN 100
80 LETA=C:LETC=D:LETL=M
90 GOSUB 160:GOTO 60
100 LETB=D:LETD=C:LETM=L
110 GOSUB 140:GOTO 60
120 LETX=(C+D)/2:PRINT'XM='X
130 GOSUB 180:PRINT'F(XM)='F:STOP
140 LETC=A+(1-K)*(B-A)
150 LETX=C:GOSUB 180:LETL=F:RETURN
160 LETD=A+K*(B-A)
170 LETX=D:GOSUB 180:LETM=F:RETURN
180 LETF=((.1*X-2)*X+10)*X
190 RETURN:END
```

мум которой вычисляется аналитически. После приближенного нахождения экстремума \bar{x}_m (максимума или минимума) можно задать $x_1 = \bar{x}_m$ и повторить поиск. Таким образом, с помощью итерационной процедуры значение x_m уточняется до получения его с заданной погрешностью ϵ . Как видно из рис. 4.8, этот метод обеспечивает поиск как максимумов, так и минимумов $F(x)$, в том числе для случая $F(x) = 0$, причем точка x_m может лежать в интервале $x_1 \pm h$ (интерполяция) и быть вне его (экстраполяция). Алгоритм реализации этого метода следующий.

1. Задаем начальное приближение x_1 для x_m и вычисляем два смежных значения аргумента $F(x)$: $x_0 = x_1 - h$ и $x_2 = x_1 + h$, где h — полуинтервал интерполяции — экстраполяции.

2. Вычисляем три значения $F(x)$: $F(x_0) = F_0$, $F(x_1) = F_1$ и $F(x_2) = F_2$.

3. Находим коэффициенты

$$c = \frac{F_0}{2h^2} - \frac{F_1}{h^2} + \frac{F_2}{2h^2} = \frac{1}{2h^2} (F_0 - 2F_1 + F_2),$$

$$b = \frac{-F_0(2x_1 + h) + 4F_1x_1 - F_2(2x_1 - h)}{2h^2}$$

параболы $y(x) = x^2 + bx + c$, проходящей через выбранные три узла интерполяции — экстраполяции $F(x)$, и по ним вычисляем аналитически положение экстремума

$$\bar{x}_m = -\frac{b}{2c} = \frac{1}{2} \frac{F_0(2x_1 + h) + 4F_1x_1 + F_2(2x_1 - h)}{F_0 - 2F_1 + F_2}$$

4. Проверяем выполнение условия $(\bar{x}_m - x_1) < \epsilon$. Если оно не выполняется, задаем $x_1 = \bar{x}_m$ и идем к п. 1. Если выполняется, считаем \bar{x}_m найденным с заданной погрешностью ϵ , вычисляем $F(x_m)$ и останавливаем счет.

Программа 4.28.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ЭКСТРЕМУМОВ ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ'
20 PRINT 'МЕТОДОМ КВАДРАТИЧНОЙ ИНТЕРПОЛЯЦИИ-ЭКСТРАПОЛЯЦИИ'
100 INPUT 'ВВЕДИТЕ ШАГ H=' H
105 INPUT 'ВВЕДИТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X1=' Z
110 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E=' E
115 LET X=Z:GOSUB 200:LET U=F
120 LET X=Z-H:GOSUB 200:LET W=F
130 LET X=Z+H:GOSUB 200:LET V=F
140 LET T=W*(2*Z+H)-4*U*Z+V*(2*Z-H)
150 LET T=T/(W-2*U+V)/2
160 IF ABS(T-Z)<E THEN 180
170 LET Z=T:GOTO 115
180 PRINT 'XМ=' T
190 PRINT 'F(XМ)=' U:STOP
200 LET F=((0.1*X-2)*X+10)*X
210 RETURN:END

```

Пример. Вычисление $F(x)$ для функции (4.22) производится подпрограммой, записанной со строки 200. Для $h=0.1$; $x_1=3$ и $\epsilon=0.001$ получим $x_m=3.33333342$ и $F(x_m)=14.8148146$, а для $x_1=8$ $x_m=9.999499962$ и $F(x_m)=2.414881321 \cdot 10^{-7}$. Значения $F(x_m)$ показывают, что в первом случае обнаружен максимум $F(x)$, а во втором — минимум,

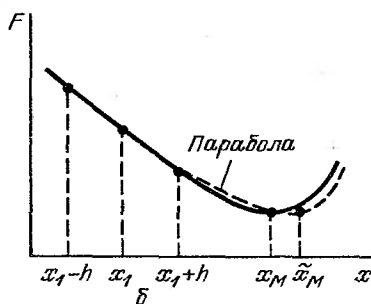
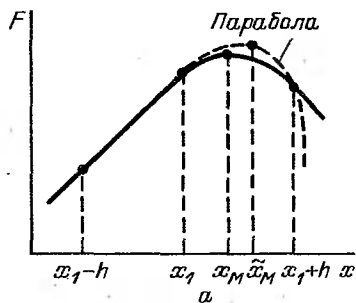


Рис. 4.8. Поиск экстремумов максимума (а) и минимума (б) методом квадратичной интерполяции — экстраполяции

причем в обоих случаях экстремумы находились за пределами интервала $x_1 \pm h$.

Сравнение методов одномерной оптимизации показывает, что для простой функции $F(x)$, например вида (4.22), они обеспечивают примерно одинаковое время поиска (около 10 с). Исключением является последний метод, имеющий время поиска примерно в 2,5 раза меньше. Последнее связано с тем, что функция (4.22) близка к квадратичной параболе из-за малости коэффициента при x^3 .

В большинстве случаев для гладких $F(x)$ метод квадратичной интерполяции — экстраполяции дает заметный выигрыш во времени вычислений. Удобно и то, что он без всякой перестройки программ обнаруживает как максимумы, так и минимумы $F(x)$, причем даже за пределами первоначально заданного интервала поиска. Преимущество метода

золотого сечения перед методами поразрядного приближения и дихотомии при простых $F(x)$ не выявляется, поскольку программная реализация первого метода сложнее и необходимо выполнение ряда вспомогательных операций. Однако при сложных $F(x)$ метод золотого сечения может давать существенный выигрыш во времени. Для поиска экстремумов пользуются также методом с числами Фибоначчи [3, 41], однако особым преимуществом перед методом золотого сечения он не обладает.

Многомерная оптимизация заключается в поиске экстремумов функции многих переменных $F(x_1, x_2, \dots, x_n)$. Методы такой оптимизации детально описаны в [3, 38, 41]. Из всего многообразия методов многомерной оптимизации ограничимся рассмотрением трех относительно простых методов поиска минимума $F(x_1, x_2, \dots, x_n)$.

Метод координатного спуска заключается в поочередном поиске минимума по координате x_1 , затем x_2 и т. д. После нахождения точки минимума по координате x_1 переходим к нахождению точки минимума по координате x_2 и т. д. Поиск ведется с одинаковым шагом, который уменьшается после нахождения всех значений $\bar{x}_{1m}, \bar{x}_{2m}, \dots, \bar{x}_{nm}$. Таким образом, алгоритм реализации этого метода подобен алгоритму метода поразрядного приближения и лишь дополняется циклом задания переменных x_1, x_2, \dots, x_n , внутри которого оценивается погрешность нахождения x_{im} для каждой переменной.

Программа 4.29.

```

10 PRINT 'МИНИМИЗАЦИЯ ФУНКЦИЙ N ПЕРЕМЕННЫХ МЕТОДОМ'
15 PRINT '      КООРДИНАТНОГО СПУСКА'
20 INPUT 'ЗАДАЙТЕ ЧИСЛО ПЕРЕМЕННЫХ N=':N:DIM A(N)
25 INPUT 'ЗАДАЙТЕ НАЧАЛЬНЫЙ ШАГ ПОИСКА H=':H:LET L=H
30 INPUT 'ЗАДАЙТЕ ТОЧНОСТЬ РЕЗУЛЬТАТА E=':E
35 FOR I=1 TO N
40 PRINT!2.0!'ВВЕДИТЕ НАЧАЛЬНЫЕ X'I':0='
45 INPUT A(I):NEXT I
50 FOR I=1 TO N:LET B=-.9E98
60 LET A(I)=A(I)+H:GOSUB 150:LET C=B:LET B=F
70 IF F-C<0 THEN 60
80 LET H=H/3:IF ABS(H)>=ABS(L/3) THEN 60
90 LET H=L:NEXT I
100 LET L=L/3:LET H=L
110 IF E/9<=L THEN 50
115 PRINT!F1.9!'F(XI MIN)=':F
120 FOR I=1 TO N
130 PRINT!2.0!'X'I' MIN=':F1.9!A(I)
135 NEXT I:PRINT 'КОНЕЦ ВЫЧИСЛЕНИЙ':STOP
140 NEXT I:PRINT 'КОНЕЦ ВЫЧИСЛЕНИЙ':STOP
150 LET F=EXP(A(1)+A(2)+A(3))/(A(1)*A(2)*A(2)*A(3)*A(3))
160 RETURN:END

```

Пример. Со строки 150 записана подпрограмма вычисления функции

$$F(x_1, x_2, x_3) = \frac{\exp(x_1 + x_2 + x_3)}{x_1 x_2^2 x_3^3} \quad (4.23)$$

трех переменных. Для начальных значений $x_{10} = x_{20} = x_{30} = 0,5$, шага $H = 0,5$ и погрешности $E = 1 \cdot 10^{-4}$ получим $x_{1m} = 0,9999745973$; $x_{2m} = 1,999974597$; $x_{3m} = 2,999979597$ и $F(x_i)_{\min} = 3,735451794$.

Время счета около 1,5 мин.

Метод спирального координатного спуска отличается от рассмотренного выше лишь тем, что шаг H меняется каждый раз при переходе от поиска минимума по одной переменной к поиску минимума по другой переменной. В трехмерном пространстве это напоминает спуск во впадину по спирали. Обычно этот метод дает некоторое сокращение времени поиска.

Программа 4.30.

Со строки 120 записана подпрограмма вычисления функции (4.23). Для $x_{10} = x_{20} = x_{30} = 0,5$; $H = 0,5$ и $E = 1 \cdot 10^{-4}$ получим $x_{1m} = 1,000032$; $x_{2m} = 2,000032$; $x_{3m} = 3,000032$ и $F(x_i)_{\min} = 3,735451795$ при $t_c \approx 1,5$ мин.

Метод координатного спуска с квадратичной интерполяцией — экстраполяцией основан на последовательном поиске минимума каждой переменной с применением для этого метода квадратичной интерполяции — экстраполяции.

Программа 4.31.

Пример. Со строки 250 записана подпрограмма вычисления функции (4.23). При $H = 0,01$; $E = 1 \cdot 10^{-4}$; $x_{10} = x_{20} = x_{30} = 0,5$ получим $x_{1m} = 1,000033409$; $x_{2m} = 2,000016758$; $x_{3m} = 3,000008995$ и $F(x_i)_{\min} = 3,735514058$ ($t_c \approx 1$ мин).

Применение многомерной оптимизации для решения систем уравнений (линейных и нелинейных). Если дана система уравнений

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0, \\ f_2(x_1, x_2, \dots, x_n) = 0, \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0, \end{cases}$$

то поиск минимума функций

$$F(x_i) = \sum_{i=1}^n |f_i(x_i)| \quad \text{или} \quad F(x_i) = \sum_{i=1}^n (f_i(x_i))^2$$

дает неизвестные этой системы.

Пример. Найти неизвестные системы из двух нелинейных уравнений

$$x_1^2 + x_2^2 = 1; \quad x_1^3 - x_2 = 0.$$


```

10 PRINT'МИНИМИЗАЦИЯ ФУНКЦИЙ N ПЕРЕМЕННЫХ МЕТОДОМ'
15 PRINT' СПИРАЛЬНОГО КООРДИНАТНОГО СПУСКА'
20 INPUT'ЗАДАЙТЕ ЧИСЛО ПЕРЕМЕННЫХ N='N:DIM A(N)
25 INPUT'ЗАДАЙТЕ НАЧАЛЬНЫЙ ШАГ ПОИСКА H='H
30 INPUT'ЗАДАЙТЕ ТОЧНОСТЬ РЕЗУЛЬТАТА E='E
35 FOR I=1 TO N
40 PRINT!2.0!'ВВЕДИТЕ НАЧАЛЬНЫЕ X'I' 0='
45 INPUT A(I):NEXT I
50 FOR I=1 TO N:LETB=.9E98
60 LETA(I)=A(I)+H:GOSUB 120:LETC=B:LETB=F
70 IF F<B THEN LETC=B
75 NEXT I
80 LETH=H/5: IF ABS(H)>E/5 THEN 50
90 PRINT!F1.9!'F(XI MIN)='F
100 FOR I=1 TO N
110 PRINT!2.0!'X'I' MIN='!F1.9! A(I)
115 NEXT I:PRINT'КОНЕЦ ВЫЧИСЛЕНИЙ':STOP
120 LETF=EXP(A(1)+A(2)+A(3))/(A(1)*A(2)*A(2)*A(3)*A(3)*A(3))
130 RETURN: END

```

Программа 4.31.

```

10 PRINT'МИНИМИЗАЦИЯ ФУНКЦИЙ N ПЕРЕМЕННЫХ F(XI) МЕТОДОМ'
15 PRINT' КВАДРАТИЧНОЙ ИНТЕРПОЛЯЦИИ-ЭКСТРАПОЛЯЦИИ'
20 INPUT'ЗАДАЙТЕ ЧИСЛО ПЕРЕМЕННЫХ N='N:DIM A(N)
25 INPUT'ЗАДАЙТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕНИЯ E='E
30 INPUT'ЗАДАЙТЕ ИНТЕРВАЛ ИНТЕРПОЛЯЦИИ H='H
35 FOR I=1 TO N
40 PRINT!2.0!'ВВЕДИТЕ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ X'I'НАЧ='
50 INPUT A(I):NEXT I
100 LETS=0
110 FOR I=1 TO N
120 LETZ=A(I):GOSUB 250:LETU=F
130 LETA(I)=Z-H:GOSUB 250:LETW=F
140 LETA(I)=Z+H:GOSUB 250:LETU=F
150 LETT=U*(2*Z+H)-4*U*Z+U*(2*Z-H)
160 LETT=T/(W-2*U+U)/2
170 IF ABS(T-Z)>E THEN LETS=1
180 LETA(I)=T:NEXT I
190 IF S<>0 THEN 100
200 PRINT!F1.9!'F(XI MIN)='F
210 FOR I=1 TO N
220 PRINT!2.0!'X'I' MIN='!F1.9! A(I)
230 NEXT I
240 PRINT'КОНЕЦ ВЫЧИСЛЕНИЙ':STOP
250 LETF=EXP(A(1)+A(2)+A(3))/(A(1)*A(2)*A(2)*A(3)*A(3)*A(3))
260 RETURN:END

```

Для них имеем

$$F(x_1, x_2) = (x_1^2 + x_2^2 - 1)^2 + (x_1^3 - x_2^2)^2.$$

Воспользовавшись программой 4.31 при $H=0,01$; $E=1 \cdot 10^{-4}$ и $x_{10}=x_{20}=1$, получим $x_{1m}=0,8259$; $x_{2m}=0,5636$ и $F(x_{1m}, x_{2m}) = 2,2726 \cdot 10^{-4}$. Подстановка x_{1m} и x_{2m} в исходные уравнения дает $x_{1m}^2 + x_{2m}^2 = 0,9998 \approx 1$ и $x_{1m}^3 - x_{2m}^2 = -2 \cdot 10^{-4} \approx 0$.

§ 4.7. Численное дифференцирование и вычисление коэффициентов чувствительности

Численное дифференцирование аналитически или таблично заданной функции $y(x)$ заключается в замене $y(x)$ интерполяционным полиномом $P(x)$, производные $d^n P(x)/dx^n \approx d^n y(x)/dx^n$ которого можно найти аналитически с помощью соответствующих фор-

мул. Для функций, заданных таблично со случайной погрешностью, точность численного дифференцирования может быть низкой.

Численное дифференцирование при равномерно расположенных узлах с интерполяцией реализуется следующими формулами (для 3, 4 и 5 узлов):

$$\begin{aligned}
 y'(x_0 + ph)_3 &= \\
 &= \frac{1}{h} [(p-0,5) y_{-1} - 2py_0 + (p+0,5) y_1], \\
 y'(x_0 + ph)_4 &= \frac{1}{h} \left(-\frac{3p^2 - 6p + 2}{6} y_{-1} + \right. \\
 &+ \frac{3p^2 - 4p - 1}{2} y_0 - \frac{3p^2 - 2p - 2}{2} y_1 + \\
 &\left. + \frac{3p^2 - 1}{6} y_2 \right),
 \end{aligned}$$

$$y'(x_0 + \rho h)_5 = \frac{1}{h} \left(\frac{2\rho^3 - 3\rho^2 - \rho}{12} y_{-2} - \frac{4\rho^3 - 3\rho^2 - 8\rho + 4}{6} y_{-1} + \frac{2\rho^3 - 5\rho}{2} y_0 - \frac{4\rho^3 + 3\rho^2 - 8\rho - 4}{6} y_1 + \frac{2\rho^3 + 3\rho^2 - \rho - 1}{12} y_2 \right).$$

В этих формулах $\rho = (x - x_0)/h$ и $x = x_0 + \rho h$. Вычисление $y'(x)$ по последней формуле реализовано программой 4.32, в которой ординаты y_{-2}, y_{-1}, y_0, y_1 и y_2 обозначены как A, B, C, D и E.

Программа 4.32.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ПЕРВОЙ ПРОИЗВОДНОЙ, ТАБЛИЧНО ЗАДАННОЙ'
20 PRINT 'N=3-5 ОРДИНАТАМИ ФУНКЦИИ ПРИ РАВНОМЕРНОМ'
30 PRINT 'РАСПОЛОЖЕНИИ УЗЛОВ ИНТЕРПОЛЯЦИИ': PRINT ''
40 INPUT 'ВВЕДИТЕ X0=' X: INPUT 'ВВЕДИТЕ ШАГ H=' H
50 INPUT 'ЗАДАЙТЕ ЧИСЛО ОРДИНАТ N=' N
60 IF N=3 GOTO 100
70 IF N=4 GOTO 110
80 IF N=5 GOTO 120
90 INPUT 'ПОВТОРИТЕ ВВОД N=' N: GOTO 60
100 INPUT 'ВВЕДИТЕ ОРДИНАТЫ Y-1, Y0, Y1' A, B, C: GOTO 130
110 INPUT 'ВВЕДИТЕ ОРДИНАТЫ Y-1, Y0, Y1, Y2' A, B, C, D: GOTO 130
120 INPUT 'ВВЕДИТЕ ОРДИНАТЫ Y-1, Y-1, Y0, Y1, Y2' A, B, C, D, E
130 INPUT 'ВВЕДИТЕ X=' X: LET F=(X-W)/H
140 IF N=4 GOTO 180
150 IF N=5 GOTO 210
160 LET F=((P-.5)*A-2*P*B+(P+.5)*C)/H
170 PRINT 'DY/DX=' F: GOTO 130
180 LET F=(-(3*P*P-6*P+2)*A/6+(3*P*P-4*P-1)*B/2)
190 LET E=(-(3*P*P-2*P-2)*C/2+(3*P*P-1)*D/6)
200 LET F=(F+E)/H: GOTO 170
210 LET F=((2*P-3)*P-1)*P+1)*A/12
220 LET F=F-(((4*P-3)*P-8)*P+4)*B/6
230 LET F=F+(2*P*P-5)*P*C/2
240 LET F=F-(((4*P+3)*P-8)*P-4)*D/6
250 LET F=F+(((2*P+3)*P-1)*P-1)*E/12
260 LET F=F/H: GOTO 170: END

```

Формулы численного дифференцирования для узлов $y(x)$ существенно проще приведенных, так как в узлах P принимает фиксированные значения. Особенно простыми являются формулы для центрального узла ($P=0$) (см. табл. 4.1). Эти формулы удобны для численного дифференцирования таблично заданных функций в точке $x=x_0$.

Частные производные функции ряда переменных $f(x_1, x_2, \dots, x_n)$ вычисляются по приведенным выше формулам, если задавать приращение одной из переменных и оставлять неизменными (равными заданным значениям) остальные переменные.

Анализ чувствительности функции $f(x_1, x_2, \dots, x_n)$ к изменению ее параметров $x_1,$

Пример. Найти производную $y'(x)$ в точке $x=0,51$ функции $y(x)$, заданной при $x_0=0,5$ и $h=0,02$ пятью значениями y_i : $y_{-2} = 0,35889029$, $y_{-1} = 0,35553253$, $y_0 = 0,35206533$, $y_1 = 0,34849251$ и $y_2 = 0,344818$. Введя эти данные, получим $y'(0,51) = -0,1786491875$.

x_2, \dots, x_n основан на вычислении абсолютного приращения функции по формуле

$$\Delta f(x_1, x_2, \dots, x_n) = \frac{\partial f}{\partial x_1} \Delta x_1 + \frac{\partial f}{\partial x_2} \Delta x_2 + \dots + \frac{\partial f}{\partial x_n} \Delta x_n.$$

Таблица 4.1

Формулы численного дифференцирования (производные в центральных узлах)

Число узлов	Производные	Погрешности $y=y_0$
3	$y' = (y_1 - y_{-1})/2h$ $y'' = (y_1 - 2y_0 + y_{-1})/h^2$	$-h^2 y'''/6$ $-h^2 y^{IV}/12$
5	$y' = (-y_2 + 8y_1 - 8y_{-1} + y_{-2})/12h$ $y'' = (-y_2 + 16y_1 - 30y_0 + 16y_{-1} - y_{-2})/12h^2$ $y''' = (y_2 - 2y_1 + 2y_{-1} - y_{-2})/2h^3$	$-h^4 y^{IV}/30$ $-h^4 y^{V}/790$ $-h^2 y^{VI}/4$
7	$y' = (y_3 - 9y_2 + 45y_1 - 45y_{-1} + 9y_{-2} - y_{-3})/60h$ $y'' = (2y_3 - 27y_2 + 270y_1 - 490y_0 + 270y_{-1} - 27y_{-2} + 2y_{-3})/180h^2$ $y''' = (-y_3 + 8y_2 - 13y_1 + 13y_{-1} - 8y_{-2} + y_{-3})/8h^3$	$-h^6 y^{VII}/140$ $-h^6 y^{VIII}/560$ $7 h^4 y^{VII}/120$

где коэффициенты $\partial f/\partial x_i$ являются абсолютными коэффициентами чувствительности к изменению параметра x_i . Применяют и относительные коэффициенты чувствительности

$$S_i = \frac{\partial f/f}{\partial x_i/x_i} = \lim_{\Delta x_i \rightarrow 0} \frac{\Delta f/f}{\Delta x_i/x_i} \quad (4.24)$$

С их помощью легко вычисляется относительное приращение функции при заданных относительных изменениях параметров (переменных) $\Delta x_i/x_{i0}$:

$$\begin{aligned} \frac{\Delta f}{f(x_{10}, x_{20}, \dots, x_{n0})} &= \\ &= S_1 \frac{\Delta x_1}{x_{10}} + S_2 \frac{\Delta x_2}{x_{20}} + \dots + S_n \frac{\Delta x_n}{x_{n0}} \end{aligned}$$

Вычисление трех частных производных и относительных коэффициентов чувствительности функции $f(x_1, x_2, \dots, x_n)$ ряда переменных можно организовать следующим образом.

1. Будем считать переменные x_1, x_2, \dots, x_n переменными массива $X(I)$ и организуем ввод числа переменных N и начальных значений переменных $x_{10}, x_{20}, \dots, x_{n0}$.

2. Вводим приращение $\Delta x = H$.

3. Организуем цикл вычислений частных производных по каждой переменной с управляющей переменной $I=1, 2, \dots, N$. Внутри цикла задаем $x_i = (x_{i0} - 2h), (x_{i0} - h), \dots, (x_{i0} + 2h)$ (для 5 узлов) и для этих значений x_i , обращаясь к подпрограмме вычисления $f(x_1, x_2, \dots, x_n)$, находим ординаты $f_{-2} = y_{-2}, f_{-1} = y_{-1}, \dots, f_2 = y_2$. После этого вычисляем производные $f'(x_{i0}) = y'(x_{i0}), f''(x_{i0}) = y''(x_{i0}), f'''(x_{i0}) = y'''(x_{i0})$ по формулам табл. 4.1 и коэффициенты S_i (4.24) и выводим их на печать.

4. После выхода из цикла идем к п 2 и повторяем вычисления (если это необходимо).

При $N=1$ по этому алгоритму вычисляются три производные и коэффициент S функции одной переменной $f(x)$.

Программа 4.33.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ТРЕХ ЧАСТНЫХ ПРОИЗВОДНЫХ И'
20 PRINT 'ОТНОСИТЕЛЬНОГО КОЭФФИЦИЕНТА ЧУВСТВИТЕЛЬНОСТИ'
30 PRINT 'ФУНКЦИИ F(X) РЯДА ПЕРЕМЕННЫХ, ЗАДАННОЙ'
40 PRINT '5 ОРДИНАТАМИ-ПОДПРОГРАММА СО СТРОКИ 400'
50 INPUT 'ВВЕДИТЕ ЧИСЛО ПЕРЕМЕННЫХ N=' N
60 DIM X(N), A(N), B(N), C(N), S(N): FOR I=1 TO N
70 PRINT !2.0! 'ВВЕДИТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X('I')'
80 INPUT X(I): NEXT I
90 INPUT 'ЗАДАЙТЕ АБСОЛЮТНОЕ ПРИРАЩЕНИЕ X(I) H=' H
100 FOR I=1 TO N: LET X(0)=X(I)
110 LET X(1)=X(0)-2*H: GOSUB 400: LET F2=F: GOSUB 400: LET F1=F
120 GOSUB 400: LET F0=F: GOSUB 400: LET E1=F: GOSUB 400: LET E2=F
130 LET X(1)=X(0): LET A(I)=(E2+8*E1-8*F1+F2)/12/H
140 LET S(I)=A(I)*X(I)/F0
150 LET B(I)=(E2+16*E1-30*F0+16*F1-F2)/12/H/H
160 LET C(I)=(E2-2*E1+2*F1-F2)/2/H/H/H: NEXT I
200 FOR I=1 TO N: PRINT !2.0! 'ДЛЯ ПЕРЕМЕННОЙ X('I')'
210 PRINT !F1.9! 'D1F/DX1=' A(I)
220 PRINT !D2F/DX2=' B(I): PRINT !D3F/DX3=' C(I)
230 PRINT 'S=' S(I): NEXT I
240 PRINT 'ЗНАЧЕНИЕ ФУНКЦИИ F(XI0)=' F0: GOTO 90
400 LET F=EXP(-X(1)^2/2)/SQR(2*PI)
410 LET X(1)=X(1)+H: RETURN: END

```

Пример. Найти три производные функции нормального распределения $\psi(x) = f(x) = e^{-x^2/2}/\sqrt{2\pi}$ в точке $x_0 = 0,5$. Подпрограмма вычисления $f(x)$ при $x=X(I)$ записывается со строки 400. Задав $H=0,001$, получим выдачу в виде

$$\begin{aligned} D1F/DX1 &= -1,760326635 \cdot 10^{-1}, \\ D2F/DX2 &= -2,640406667 \cdot 10^{-1}, \\ D3F/DX3 &= 4,845 \cdot 10^{-1}, \\ S &= -2,500000002 \cdot 10^{-1}, \\ F(XI0) &= 3,3520653268 \cdot 10^{-1}. \end{aligned}$$

Точные значения производных и функции $f(x_0)$:

$$\begin{aligned} \frac{df}{dx} &= -0,1760326633, \\ \frac{d^2f}{dx^2} &= -0,2640489951, \\ \frac{d^3f}{dx^3} &= 0,48408982, \end{aligned}$$

$$f(x_0) = 0,3520653267.$$

Если значения $x_{10}, x_{20}, \dots, x_{n0}$ сильно различаются, то для повышения точности численного дифференцирования целесообразно задавать не абсолютные приращения $\Delta x_i = h$, а относительные $p = \Delta x_i/x_{i0}$. Тогда при вычислениях п. 3 в описанном выше алгоритме нужно задавать абсциссы $(x_{i0} - 2px_{i0}), (x_{i0} - px_{i0}), \dots, (x_{i0} + 2px_{i0})$. Кроме того, в этом случае

$$S_i = (\partial f_i / \partial x_i) / p.$$

Программа 4.34.

Работу этой программы можно проверить по примеру, приведенному выше для программы 4.33 (при $p=0,002$ получим аналогичные результаты).

Подобные программы могут быть составлены и для численного дифференцирования аналитически заданных функций по 7 точкам.

Программа 4.34.

```

10 PRINT'ВЫЧИСЛЕНИЕ ТРЕХ ЧАСТНЫХ ПРОИЗВОДНЫХ И'
20 PRINT'ОТНОСИТЕЛЬНОГО КОЭФФИЦИЕНТА ЧУВСТВИТЕЛЬНОСТИ'
30 PRINT'ФУНКЦИИ F(XI) РЯДА ПЕРЕМЕННЫХ, ЗАДАННОЙ'
40 PRINT'АНАЛИТИЧЕСКИ-ПОДПРОГРАММА СО СТРОКИ 400'
45 PRINT'ПРИ ЗАДАНИИ ОТНОСИТЕЛЬНОГО ИЗМЕНЕНИЯ ПЕРЕМЕННЫХ'
50 INPUT'ВВЕДИТЕ ЧИСЛО ПЕРЕМЕННЫХ N='N
60 DIM X(N),A(N),B(N),C(N),S(N):FOR I=1 TO N
70 PRINT!2.0!'ВВЕДИТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X('I)')
80 INPUT X(I): NEXT I
90 INPUT'ЗАДАЙТЕ ОТНОСИТЕЛЬНОЕ ПРИРАЩЕНИЕ X(I) P='P
100 FOR I=1 TO N:LETX(0)=X(I):LETH=P*X(0)
110 LETX(I)=X(0)-2*N:H:GOSUB 400:LETF2=F:GOSUB 400:LETF1=F
120 GOSUB 400:LETF0=F:GOSUB 400:LETE1=F:GOSUB 400:LETE2=F
130 LETX(I)=X(0):LETA(I)=(-E2+8*E1-8*F1+F2)/12/H
140 LETS(I)=A(I)*X(I)/F0
150 LETB(I)=(-E2+16*E1-30*F0+16*F1-F2)/12/H/H
160 LETC(I)=(E2-2*E1+2*F1-F2)/2/H/H/H:NEXT I
200 FOR I=1 TO N:PRINT!2.0!'ДЛЯ ПЕРЕМЕННОЙ X('I)')
210 PRINT!F1.9!'D1F/DX1='A(I)
220 PRINT'D2F/DX2='B(I):PRINT'D3F/DX3='C(I)
230 PRINT'S='S(I):NEXT I
240 PRINT'ЗНАЧЕНИЕ ФУНКЦИИ F(XI0)='F0:GOTO 90
400 LETF=EXP(-X(1)^2/2)/SGR(2*#PI)
410 LETX(I)=X(I)+H:RETURN:END

```

Программа 4.35.

```

10 PRINT'ВЫЧИСЛЕНИЕ ТРЕХ ЧАСТНЫХ ПРОИЗВОДНЫХ И'
20 PRINT'ОТНОСИТЕЛЬНОГО КОЭФФИЦИЕНТА ЧУВСТВИТЕЛЬНОСТИ'
30 PRINT'ФУНКЦИИ F(XI) РЯДА ПЕРЕМЕННЫХ, ЗАДАННОЙ'
40 PRINT'АНАЛИТИЧЕСКИ-ПОДПРОГРАММА СО СТРОКИ 400'
50 INPUT'ВВЕДИТЕ ЧИСЛО ПЕРЕМЕННЫХ N='N
60 DIM X(N),A(N),B(N),C(N),S(N):FOR I=1 TO N
70 PRINT!2.0!'ВВЕДИТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X('I)')
80 INPUT X(I): NEXT I
90 INPUT'ЗАДАЙТЕ ОТНОСИТЕЛЬНОЕ ПРИРАЩЕНИЕ X(I) P='P
100 FOR I=1 TO N:LETX(0)=X(I):LETH=X(0)*P
110 LETX(I)=X(0)-3*N:H:GOSUB 400:LETF3=F
120 GOSUB 400:LETF2=F:GOSUB 400:LETF1=F:GOSUB 400:LETE0=F
130 GOSUB 400:LETE1=F:GOSUB 400:LETE2=F:GOSUB 400:LETE3=F
140 LEТА(I)=(E3-9*E2+45*E1-45*F1+9*F2-F3)/60/H
150 LETB(I)=2*E3-27*E2+27*E1-490*E0
160 LETC(I)=(B(I)+270*F1-27*F2+2*F3)/180/(H*N)
170 LETC(I)=(8*E2-E3-13*E1+13*F1-8*F2+F3)/8/(H^3)
180 LETS(I)=A(I)*X(0)/E0:LETX(I)=X(0):NEXT I
200 FOR I=1 TO N:PRINT!2.0!'ДЛЯ ПЕРЕМЕННОЙ X('I)')
210 PRINT!F1.9!'D1F/DX1='A(I)
220 PRINT'D2F/DX2='B(I):PRINT'D3F/DX3='C(I)
230 PRINT'S='S(I):NEXT I
240 PRINT'ЗНАЧЕНИЕ ФУНКЦИИ F(XI0)='E0:GOTO 90
400 LETF=EXP(-X(1)^2/2)/SGR(2*#PI)
410 LETX(I)=X(I)+H:RETURN:END

```

Пример. Используя данные програм-
мы 4.33, получим

$$\begin{aligned}
 D1F/DX1 &= -1,760326636 \cdot 10^{-1}, \\
 D2F/DX2 &= -2,640489378 \cdot 10^{-1}, \\
 D3F/DX3 &= 4,840943756 \cdot 10^{-1},
 \end{aligned}$$

$$\begin{aligned}
 S &= -2,500000004 \cdot 10^{-1}, \\
 F(XI0) &= 3,3520653268 \cdot 10^{-1}.
 \end{aligned}$$

В этой программе $H = \text{const}$. В приведенной ниже программе задается $P = \Delta x_i / x_{i0} = \text{const}$.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ТРЕХ ЧАСТНЫХ ПРОИЗВОДНЫХ И'
20 PRINT 'ОТНОСИТЕЛЬНОГО КОЭФФИЦИЕНТА ЧУВСТВИТЕЛЬНОСТИ'
30 PRINT 'ФУНКЦИИ F(XI) РЯДА ПЕРЕМЕННЫХ, ЗАДАННОЙ'
40 PRINT 'АНАЛИТИЧЕСКИ-ПОДПРОГРАММА СО СТРОКИ 400'
50 INPUT 'ВВЕДИТЕ ЧИСЛО ПЕРЕМЕННЫХ N=';N
60 DIM X(N),A(N),B(N),C(N),S(N):FOR I=1 TO N
70 PRINT!2.0!'ВВЕДИТЕ НАЧАЛЬНОЕ ЗНАЧЕНИЕ X('I')'
80 INPUT X(I):NEXT I
90 INPUT 'ЗАДАЙТЕ АБСОЛЮТНОЕ ПРИРАЩЕНИЕ X(I) H=';H
100 FOR I=1 TO N:LETX(0)=X(I)
110 LETX(I)=X(0)+3*H:GOSUB 400:LETF3=F
120 GOSUB 400:LETF2=F:GOSUB 400:LETF1=F:GOSUB 400:LETE0=F
130 GOSUB 400:LETE1=F:GOSUB 400:LETE2=F:GOSUB 400:LETE3=F
140 LETA(I)=(E3-9*E2+45*E1-45*F1+9*F2-F3)/60/H
150 LETB(I)=2*E3-27*E2+270*E1-490*E0
160 LETC(I)=(B(I)+270*F1-27*F2+2*F3)/180/(H*H)
170 LETS(I)=(8*E2-E3-13*E1+13*F1-8*F2+F3)/8/(H^3)
180 LETS(I)=A(I)*X(0)/E0:LETX(I)=X(0):NEXT I
200 FOR I=1 TO N:PRINT!2.0!'ДЛЯ ПЕРЕМЕННОЙ X('I')'
210 PRINT!F1.9!'D1F/DX1=';A(I)
220 PRINT!D2F/DX2=';B(I):PRINT!D3F/DX3=';C(I)
230 PRINT!S=';S(I):NEXT I
240 PRINT 'ЗНАЧЕНИЕ ФУНКЦИИ F(XI0)=';E0:GOTO 90
400 LETF=EXP(-X(1)^2/2)/SOR(2*#PI)
410 LETX(I)=X(I)+H:RETURN:END
    
```

Работу этой программы также можно проверить по приведенному выше примеру. Следует отметить, что из-за ошибок округления точность численного дифференцирования при малых h и p при 7 точках улучшается незначительно, а иногда даже ухудшается (в сравнении с дифференцированием по 5 точкам).

§ 4.8. Вычисление определенных интегралов

Определенный интеграл

$$I = \int_a^b f(x) dx \quad (4.25)$$

с пределами интегрирования a и b можно трактовать как площадь фигуры (рис. 4.9), ограниченной ординатами a и b , осью абсцисс x и графиком подынтегральной функции $f(x)$.

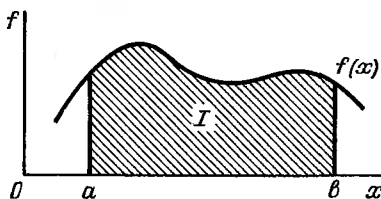


Рис. 4.9. Графическая иллюстрация численного интегрирования

Обыкновенный определенный интеграл, у которого известна его первообразная $F(x)$, вычисляется по формуле Ньютона — Лейбница

$$I = F(b) - F(a).$$

Поэтому достаточно вычислить значения функции $F(x)$. Большое количество определенных интегралов приведено в книге [30].

Численное интегрирование применяется, если нахождение $F(x)$ сложно или невозможно. Оно заключается в интерполяции $f(x)$ на отрезке $[a, b]$ подходящим полиномом, для которого определенный интеграл вычисляется по формулам численного интегрирования. Обычно отрезок $[a, b]$ разбивается на m частей, к каждой из которых применяется соответствующая простая формула. Таким образом получают составные (или сложные) формулы численного интегрирования.

Метод прямоугольников — простейший прием численного интегрирования, при котором функция $y = f(x)$ заменяется интерполяционным многочленом нулевого порядка. Для повышения точности интегрирования отрезок $[a, b]$ разбивается на m частей и формула прямоугольника применяется к каждому отрезку. Приведем обобщенную формулу прямоугольников:

$$I \approx \frac{b-a}{m} \sum_{i=0}^{m-1} y_i.$$

Вследствие низкой точности этот метод почти не применяется.

Модифицированный метод прямоугольников базируется на представлении $f(x)$ ординатами, смещенными на величину $0,5h$, где $h = (b-a)/m$:

$$I \approx h \sum_{i=0}^{m-1} f(x_i + 0,5h) + \frac{h^2(b-a)}{24} f''(\xi),$$

где $f''(\xi)$ — значение второй производной $f(x)$ в точке $x = \xi$, где она максимальна.

Метод трапеций заключается в линейной аппроксимации $f(x)$ на отрезке $[a, b]$. Для уменьшения погрешности $[a, b]$ разбивается

на m частей длины $h=(b-a)/m$ (при методе прямоугольников $f(x_i)=\text{const}=y_i$ на этом отрезке).

С учетом суммирования смежных ординат внутри отрезка $[a, b]$ обобщенная формула метода трапеций имеет вид

$$I \approx \frac{b-a}{m} \left(\frac{y_0}{2} + y_1 + y_2 + \dots + y_{m-1} + \frac{y_m}{2} \right) - \frac{h^2(b-a)}{12} f''(\xi).$$

В связи с легкостью программной реализации и низкой точностью рассмотренных методов программы для них не приводятся.

Метод Ньютона — Котеса основан на интерполяции $f(x)$ в n промежуточных полиноме Лагранжа. В общем случае $f(x)$ должна задаваться $(n+1)$ ординатами. Формулы интегрирования точны, если $f(x)$ — многочлен n -й степени. При $n=1$ получаем метод трапеций.

Метод Симпсона (парабол) — частный случай метода Ньютона — Котеса при $n=2$. При разбиении отрезка $[a, b]$ на m равных отрезков получим обобщенную формулу Симпсона

$$I = \int_a^b f(x) dx \approx \frac{h}{3} \{ f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \dots + 4f(b-h) + f(b) \} - \frac{mh^5}{90} f^{(4)}(\xi).$$

Выражение для остаточного члена показывает, что формула Симпсона точна, даже если $f(x)$ — многочлен третьей степени. Эта частная особенность формулы Симпсона объясняет ее преимущественное применение — у некоторых ЭВМ вычисление по ней реализовано микропрограммно.

Алгоритм реализации метода Симпсона дан на рис. 4.10.

Программа 4.37.

Пр и м е р. Вычислить интеграл

$$I = \int_0^1 \sqrt{2x+1} dx = 1,398717474 \quad (4.26)$$

при $a=0$ и $b=1$. Вычисление подынтегральной функции оформляется подпрограммой — см. строку 110. Для $m=M=8$ получим $I=1,39871724$ при $t_c \approx 5$ с.

Программа 4.37.

```

10 PRINT 'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МЕТОДОМ СИМПСОНА'
20 INPUT 'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A=' A
25 INPUT 'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B=' B
30 INPUT 'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ ИНТЕГРИРОВАНИЯ M=' M
40 LET H=(B-A)/M/2 : LET X=A
50 GOSUB 110 : LET I=F : LET N=0
60 LET X=X+H : GOSUB 110 : LET I=I+4*F
70 LET N=N+2 : IF N=2*M THEN 90
80 LET X=X+H : GOSUB 110 : LET I=I+2*F : GOTO 60
90 LET X=B : GOSUB 110 : LET I=(I+F)*H/3
95 PRINT 'ДЛЯ A=' A ' B=' B ' И M=' M: PRINT '
100 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' I: GOTO 30
110 LET F=SGR(2*X+1) : RETURN : END
    
```

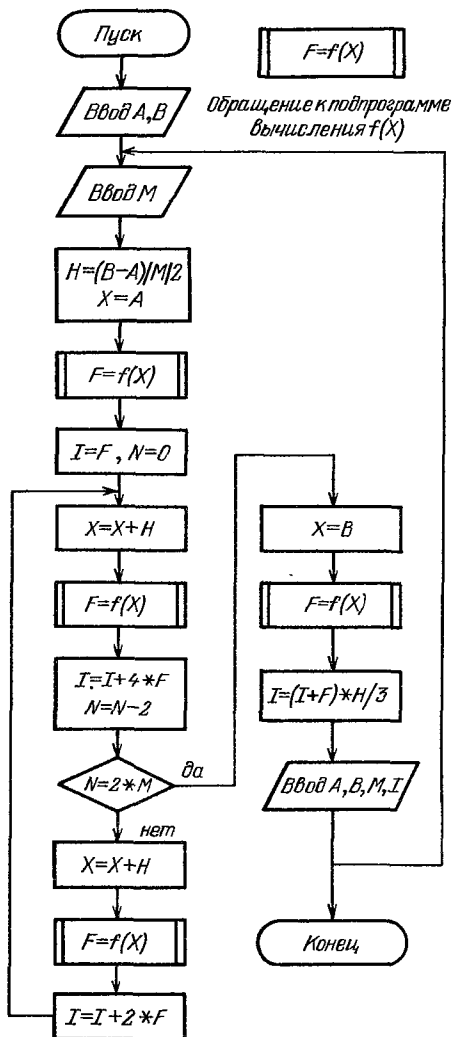


Рис. 4.10. Алгоритм численного интегрирования по составной формуле Симпсона

Интегрирование по формуле Бодэ

$$I = \int_{x_0}^{x_4} f(x) dx = \frac{2h}{45} (7y_0 + 32y_1 + 12y_2 + 32y_3 + 7y_4) - \frac{8h^7}{945} f^{(6)}(\xi),$$

примененной к каждому из m отрезков разбиения $[a, b]$, реализует следующая программа.

Программа 4.38.

```

10 PRINT 'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ ПО ФОРМУЛЕ БОДЭ'
20 INPUT 'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A=' A
30 INPUT 'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B=' B
40 INPUT 'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ ИНТЕГРИРОВАНИЯ M=' M
50 LET H=(B-A)/M: LET E=H/4: LET X=A: GOSUB 130: LET I=7*F
60 FOR N=1 TO M: LET X=X+E
70 GOSUB 130: LET I=I+32*F: LET X=X+E
80 GOSUB 130: LET I=I+12*F: LET X=X+E
90 GOSUB 130: LET I=I+32*F: LET X=X+E
100 GOSUB 130: LET I=I+14*F
110 NEXT N
120 LET I=I-7*F: LET I=I*2*E/45
125 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' I: GOTO 40
130 LET F=SQR(2*X+1): RETURN: END
    
```

Пример. Для интеграла (4.26) при $m=4$ получим $I=1,398717463$ при $t_c \approx 5$ с. Подпрограмма вычисления $f(x)$ записывается со строки 130.

Метод Уэддла базируется на применении к каждому из m отрезков разбиения $[a, b]$ формулы

$$I = \int_{x_0}^{x_6} f(x) dx = \frac{3h}{10} (y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + y_6) + \frac{47h^7}{12700} f^{VI}(\xi).$$

Программа 4.39.

Пример. Для интеграла (4.26) при $m=4$ получим $I=1,398717474$ при $t_c \approx 5$ с. Подпрограмма вычисления $f(x)$ записывается со строки 170.

Формула Ньютона — Котеса с повышенной (в сравнении с формулой Уэддла) точностью для $m=6$

$$I = \int_{x_0}^{x_6} f(x) dx \approx \frac{h}{140} (41y_0 + 216y_1 + 27y_2 + 272y_3 + 27y_4 + 216y_5 + 41y_6) - \frac{9h^9}{1400} f^{VIII}(\xi)$$

реализована в следующей программе.

Программа 4.40.

Пример. Для интеграла (4.26) при $m=4$ получим $I=1,398717474$ при $t_c \approx 5$ с. Подпрограмма вычисления $f(x)$ записывается со строки 130.

Приведенные выше методы интегрирования имеют равномерное расположение абсцисс x_i , для которых вычисляются $y_i = f(x_i)$. При этом крайние значения x_i равны a и b (за исключением модифицированного метода прямоугольников).

Метод Чебышева основан на вычислении значения определенного интеграла

$$I = \int_{-1}^{+1} f(t) dt \approx \frac{2}{n} \sum_{i=1}^n f(t_i) = b_i \sum_{i=1}^n f(t_i) \quad (4.27)$$

по формуле, содержащей одинаковые множители $b_i = 2/n$ перед $f(t_i)$. Однако при этом абсциссы t_i располагаются неравномерно. Так, для $n=3$ $t_1 = -\sqrt{2}/2$, $t_2 = 0$ и $t_3 = \sqrt{2}/2$

интеграл (4.27) приводится к виду (4.25) подстановкой

$$x = \frac{b+a}{2} + \frac{b-a}{2} t. \quad (4.28)$$

Формула (4.27) точна для полиномов степени n .

Программа 4.41.

Пример. Для интеграла (4.26) при $m=8$ получим $I=1,398717533$ при $t_c \approx 6$ с. Вычисление $f(x)$ оформлено подпрограммой (строка 140).

Метод Гаусса основан на интерполяции $f(x)$ полиномом Лагранжа, но абсциссы x_i выбираются из условия обеспечения минимума погрешности интерполяции. При этом интеграл (4.25) подстановкой (4.28) сводится к виду

$$I = \int_{-1}^{+1} f(t) dt = \sum_{i=1}^n A_i f(t_i). \quad (4.29)$$

Метод Гаусса обычно обеспечивает повышенную точность — формула (4.29) верна для полиномов до $(2n-1)$ -й степени. Для $n=3$ $A_1 = 5/9$, $t_1 = -\sqrt{1/3}$, $A_2 = 8/9$, $t_2 = 0$, $A_3 = 5/9$ и $t_3 = \sqrt{1/3}$. Остаточный член (погрешность) при этом равен

$$R_3 = \frac{1}{15750} \left(\frac{b-a}{2} \right)^7 f^{VI}(\xi).$$

Для повышения точности интегрирования отрезок $[a, b]$ дробится на m частей.

Программа 4.42.

Пример. Для интеграла (4.26) при $M=8$ получим $I=1,398717474$ при $t_c \approx 6$ с. Подпрограмма вычисления $f(x)$ записана со строки 130. При численном интегрировании с заданной точностью оценка погрешности интегрирования обычно сложна и требует вычисления высших производных $f(x)$. Поэтому на практике принято проводить интегрирование для ряда $m=2, 4, 8$ и т. д., считая верным число совпадающих первых цифр результата. Этот процесс легко автоматизировать, что положено в основу программы 4.43.

Программа 4.39.

```

10 PRINT 'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МЕТОДОМ УДАЛЯ'
20 INPUT 'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A=' Q
30 INPUT 'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B=' B
40 INPUT 'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ ИНТЕГРИРОВАНИЯ M=' M
50 LET A=Q:LETH=(B-A)/M:LETX=A:GOSUB 170
55 LETI=F:LETL=H/6
60 FOR E=1 TO M
70 LETX=X+L:GOSUB 170:LETI=I+5*F
80 LETX=X+L:GOSUB 170:LETI=I+F
90 LETX=X+L:GOSUB 170:LETI=I+6*F
100 LETX=X+L:GOSUB 170:LETI=I+F
110 LETX=X+L:GOSUB 170:LETI=I+5*F
120 LETX=X+L:GOSUB 170:LETI=I+2*F
130 LETA=X: NEXT E
140 LETI=(I-F)*3*L/10
150 PRINT 'ДЛЯ A=' Q 'B=' B 'И M=' M
160 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' I:GOTO 40
170 LET F=SQR(1+2*X):RETURN:END

```

Программа 4.40.

```

10 PRINT 'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МЕТОДОМ НЬЮТОНА-КОТЕСА'
20 INPUT 'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A=' A
25 INPUT 'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B=' B
30 INPUT 'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ ИНТЕГРИРОВАНИЯ M=' M
35 LETH=(B-A)/M:LETE=H/6
40 LETX=A:GOSUB 130:LETI=41*F
50 FOR N=1 TO M:LETX=X+E
60 GOSUB 130:LETI=I+216*F:LETX=X+E
70 GOSUB 130:LETI=I+27*F:LETX=X+E
80 GOSUB 130:LETI=I+272*F:LETX=X+E
90 GOSUB 130:LETI=I+27*F:LETX=X+E
100 GOSUB 130:LETI=I+216*F:LETX=X+E
110 GOSUB 130:LETI=I+82*F:NEXT N
120 LETI=(I-41*F)*E/140
125 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' I:GOTO 30
130 LET F=SQR(2*X+1):RETURN:END

```

Программа 4.41.

```

10 PRINT 'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МЕТОДОМ ЧЕБЫШЕВА'
20 INPUT 'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A=' Q
30 INPUT 'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B=' B
40 INPUT 'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ ИНТЕГРИРОВАНИЯ M=' M
50 LET A=Q:LETT=SQR(2)/2:LETI=0:LETH=(B-A)/M
60 FOR E=1 TO M:LETW=A+H:LETC=(W+A)/2
70 LETD=(W-A)/2:LETX=C-D*T:GOSUB 120
80 LETI=I+F:LETX=C:GOSUB 120
90 LETI=I+F:LETX=X+D*T:GOSUB 120
100 LETI=I+F:LETA=W:NEXT E
110 LETI=I*H/3:PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' I:GOTO 40
120 LET F=SQR(2*X+1):RETURN:END

```

Программа 4.42.

```

05 PRINT 'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МЕТОДОМ ГАУССА'
10 INPUT 'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A=' Q
12 INPUT 'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B=' B
15 INPUT 'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ ИНТЕГРИРОВАНИЯ M=' M:LETA=Q
20 LETT=SQR(0.6):LETI=0:LETH=(B-A)/M
30 FOR E=1 TO M:LETB1=A+H
40 LETC=(B1+A)/2:LETD=(B1-A)/2
50 LETX=C-D*T:GOSUB 130
60 LETY=D*5*F/9:LETI=I+Y
70 LETX=C:GOSUB 130
80 LETY=D*8*F/9:LETI=I+Y
90 LETX=C+D*T:GOSUB 130
100 LETY=D*5*F/9:LETI=I+Y
110 LEA=B1:NEXT E
115 PRINT 'ДЛЯ A=' Q 'B=' B 'И M=' M:PRINT''
120 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' I:GOTO 15
130 LET F=SQR(2*X+1):RETURN:END

```



```

10 PRINT 'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ С ЗАДАННОЙ ТОЧНОСТЬЮ'
15 PRINT ' МЕТОДОМ ГАУССА'
20 INPUT 'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A' Q
22 INPUT 'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B' Q
25 INPUT 'ВВЕДИТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E' = 'N
30 LET N=N*60: LET M=1: LET K=0
40 LET M=2*M: LET A=Q
45 LET T=SQR(.6): LET I=0: LET H=(B-A)/M
50 FOR J=1 TO M: LET W=A+H: LET C=(W+A)/2
55 LET D=(W-A)/2: LET E=D*5/9
60 LET L=D*8/9: LET D=D*T
70 LET X=C-D: GOSUB 150: LET I=I+E*F
80 LET X=C: GOSUB 150: LET I=I+L*F
90 LET X=C+D: GOSUB 150: LET I=I+E*F
100 LET A=W: NEXT J
110 LET L=K: LET K=I
120 IF ABS(I-L)>N THEN 40
130 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I=' I: STOP
150 LET F=SQR(2*X+1): RETURN: END

```

Для погрешности $\varepsilon = 1 \cdot 10^{-6}$ вычисление интеграла (4.26) дает $I = 1,398717485$ при $t_c \approx 8$ с. Подпрограмма вычисления $f(x)$ записывается со строки 150.

§ 4.9. Вычисление определенных интегралов специального вида

Несобственными интегралами называются интегралы, у которых $f(x) \rightarrow \infty$ на отрезке $[a, b]$ либо $f(a) \rightarrow \infty$ или $f(b) \rightarrow \infty$.

Вычисление несобственных интегралов с $f(x) \rightarrow \infty$ на концах отрезка $[a, b]$ в принципе осуществляется методами Гаусса и Чебышева, так как у них в ходе вычислений узлы интерполяции $f(x)$ не совпадают с пределами интегрирования. Наличие особенности может повлиять на точность интегрирования.

Методы Ньютона — Котеса (в частности, метод Симпсона) удобно применять, если $f(x_0) \rightarrow \infty$ при x_0 , принадлежащем интервалу (a, b) . Равномерное расположение узлов интерполяции у этих методов облегчает контроль за их положением и исключение в ходе вычислений значений $x_i = x_0$. Если $f(x)$ имеет на концах предела интегрируемости устраняемую особенность, например неопределенность $0/0$ или ∞/∞ , можно использовать эти методы с отдельным вводом значений $f(a)$ и $f(b)$.

Программа 4.44.

Пример. При $\varepsilon = 1 \cdot 10^{-4}$ получим для интеграла (4.26) значение $I = 1,3987$ при $t_c \approx 4$ с.

Сходящиеся несобственные интегралы с бесконечными пределами интегрирования вычисляются различными приемами [4]. Так, интеграл вида

$$I = \int_a^{\infty} f(x) dx = \int_a^b f(x) dx + \int_b^{\infty} f(x) dx \quad (4.30)$$

может вычисляться как обыкновенный при выборе вспомогательного промежуточного предела интегрирования из условия

$$I = \int_b^{\infty} f(x) dx < \varepsilon.$$

где ε — заданная погрешность интегрирования. Другой прием заключается в автоматическом изменении b , например как $b_0, 2b_0, 4b_0, 8b_0$ и т. д., где b_0 — начальное значение b с контролем на каждой i -й стадии условия

$$|I_i - I_{i-1}| < \varepsilon = E.$$

При выполнении этого условия счет прекращается. Последний способ реализован в программе 4.45, где вычисляются несобственные интегралы вида (4.30) методом Гаусса.

Программа 4.45.

Пример. Для функции $f(x) = xe^{-x}$, $a = 0$ при $E = 1 \cdot 10^{-3}$ и $b_0 = 10$ получим $I = 1,000457964$ при $t_c \approx 20$ с. Отметим, что при малых E время счета может резко возрастать, а из-за погрешностей вычисления I_i и I_{i-1} программа может заикаться. В строке 150 записана подпрограмма, вычисляющая $f(x)$.

Двойные (кратные) интегралы вида

$$I = \int_a^b \int_c^d f(x, y) dx dy$$

могут вычисляться по кубатурной формуле Симпсона. При этом область интегрирования в плоскости x, y разбивается на четыре квадрата (рис. 4.11, а), а значение I вычисляется по формуле [7]

$$I \approx \frac{hk}{9} \{ [f(x_0, y_0) + f(x_2, y_0) + f(x_0, y_2) + f(x_2, y_2)] + 4 [f(x_1, y_0) + f(x_0, y_1) + f(x_2, y_1) + f(x_1, y_2)] + 16f(x_1, y_1) \}$$

Программа 4.46.

Пример. Для интеграла (подпрограмма записана со строки 140)

$$I = \int_4^{4,4} \int_2^{2,6} \frac{dx dy}{xy} = \ln 3 \cdot \ln 1,1 = 0,02500598522, \quad (4.31)$$

введя $a=4, b=4,4, c=2$ и $d=2,6$, получим $I = 0,02500698577$. Для повышения точности область интегрирования можно разбить на

Программа 4.44.

```

10 PRINT'ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ МЕТОДОМ СИМПСОНА С ЗАДАННОЙ'
15 PRINT'ТОЧНОСТЬЮ E И F(X) С ОСОБЕННОСТЯМИ ПРИ X=A И X=B'
20 PRINT': INPUT'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A='A
25 INPUT'ВВЕДИТЕ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ B='B
30 INPUT'ВВЕДИТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E='E: LETM=1
40 PRINT'ВВЕДИТЕ 0 ЕСЛИ НЕТ ОСОБЕННОСТИ ПРИ X=A'
45 INPUT'И 1 ЕСЛИ ЕСТЬ ОСОБЕННОСТЬ ПРИ X=A 'S
50 IF S=0 THEN 65
60 INPUT'ВВЕДИТЕ ЗНАЧЕНИЕ F(A)='C: 60 TO 70
65 LETX=A: 60 SUB 210: LETC=F
70 PRINT'ВВЕДИТЕ 0 ЕСЛИ НЕТ ОСОБЕННОСТИ ПРИ X=B'
75 INPUT'И 1 ЕСЛИ ЕСТЬ ОСОБЕННОСТЬ ПРИ X=B 'S
80 IF S=0 THEN 100
90 INPUT'ВВЕДИТЕ ЗНАЧЕНИЕ F(B)='D: 60 TO 110
100 LETX=B: 60 SUB 210: LETI=F: LETJ=0
110 LETM=M*2: LETH=(B-A)/M/2
120 LETN=0: LETI=0: LETX=A
130 LETX=X+H: 60 SUB 210
140 LETI=I+4*F: LETN=N+2
150 IF N=2*M THEN 180
160 LETX=X+H: 60 SUB 210
170 LETI=I+2*F: 60 TO 130
180 LETI=(I+C+D)*H/3: LETK=J: LETJ=I
190 IF ABS(I-K)>E*15 THEN 110
200 PRINT': PRINT'ЗНАЧЕНИЕ ИНТЕГРАЛА I='I: STOP
205 REM'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(X)'
210 LETF=SIN(X)/X: RETURN: END

```

Программа 4.45.

```

05 PRINT'ВЫЧИСЛ. НЕСОБСТВ. ИНТЕГРАЛА МЕТОДОМ ГАУССА'
10 INPUT'ВВЕДИТЕ НИЖНИЙ ПРЕДЕЛ ИНТЕГРИРОВАНИЯ A='0
15 INPUT'ВВЕДИТЕ НАЧАЛЬНЫЙ ВЕРХНИЙ ПРЕДЕЛ ИНТЕГР. B0='Z
20 INPUT'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА E='N
25 LETM=1: LETK=0: LETB=Z/2: LETT=SQR(.6): LETU=0
30 LETB=B*2
35 LETM=M*2: LETA=0: LETI=0: LETH=(B-A)/M
40 FOR J=1 TO M: LETC=A+H/2: LETI=H/2
50 LETE=I*5/9: LETL=I*8/9: LETD=I*T
60 LETX=C-D: 60 SUB 150: LETI=I+E*F
70 LETX=C: 60 SUB 150: LETI=I+L*F
80 LETX=C+D: 60 SUB 150: LETI=I+E*F
90 LETA=A+H: NEXT J
100 LETL=K: LETK=I
110 IF ABS(I-L)>N*30 THEN 35
120 LETV=U: LETU=I
130 IF ABS(I-U)>N/2 THEN 30
140 PRINT'ЗНАЧЕНИЕ ИНТЕГРАЛА I='I: STOP
150 LETF=X*EXP(-X): RETURN: END

```

Программа 4.46.

```

10 PRINT'ВЫЧИСЛЕНИЕ ДВОЙНОГО ИНТЕГРАЛА ПО ПРОСТОЙ'
15 PRINT' ФОРМУЛЕ СИМПСОНА': LETS=0
20 INPUT'ВВЕДИТЕ ПРЕДЕЛЫ A, B, C, D 'A, B, C, D
30 LETH=(B-A)/2: LETK=(D-C)/2
40 LETX=A: LETY=C: 60 SUB 140: LETS=S+F
50 LETX=B: 60 SUB 140: LETS=S+F
60 LETX=A: LETY=D: 60 SUB 140: LETS=S+F
70 LETX=B: 60 SUB 140: LETS=S+F
80 LETX=A+H: LETY=C: 60 SUB 140: LETS=S+4*F
90 LETY=D: 60 SUB 140: LETS=S+4*F
100 LETX=A: LETY=C+K: 60 SUB 140: LETS=S+4*F
110 LETX=B: 60 SUB 140: LETS=S+4*F
120 LETX=A+H: 60 SUB 140: LETS=S+16*F
130 PRINT'ЗНАЧЕНИЕ ИНТЕГРАЛА I='S*M*H*K/9: 60 TO 20
140 LETF=1/X/Y: RETURN: END

```

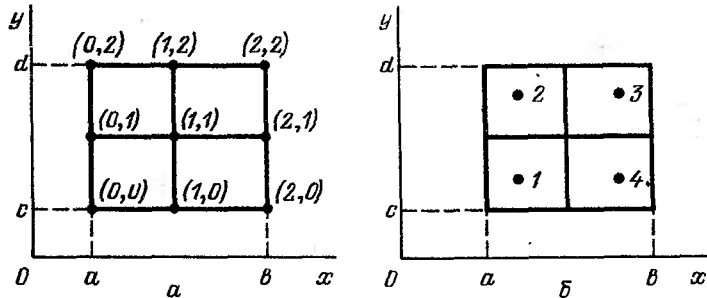


Рис. 4.11. Расположение узлов при вычислении двойных определенных интегралов методом Симпсона (а) и Гаусса (б)

несколько квадратов, вычислив I_i для каждого i -го квадрата. Программа 4.46 предусматривает последовательное суммирование значений I_i . Например, разбив область интегрирования (4.31) на 4 квадрата ($a_1=4; b_1=4,2; c_1=2,3; d_1=2,6; a_2=4,2; b_2=4,4; c_2=2,3; d_2=2,6; a_3=4,2; b_3=4,4; c_3=2; d_3=2,3; a_4=4; b_4=4,2; c_4=2$ и $d_4=2,3$ — см. рис. 4.11, б), получим

$$I = 0,02500605059.$$

Время счета для каждого квадрата менее 2 с. Кубатурная формула Гаусса с $n=2$ имеет вид

$$I = \int_a^b \int_c^d f(x, y) = PR [f(x_1, y_1) + f(x_2, y_2) + f(x_3, y_3) + f(x_4, y_4)], \quad (4.32)$$

где $P = (b-a)/2$, $R = (d-c)/2$, а узлы интерполяции расположены на расстоянии $\pm KP$ и $\pm KR$ от осей, где $K = \sqrt{1/3}$.

Программа 4.47.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ДВОЙНОГО ИНТЕГРАЛА ПО ПРОСТОЙ'
15 PRINT '    ФОРМУЛЕ ГАУССА': LETS=0
20 INPUT 'ВВЕДИТЕ ПРЕДЕЛЫ А, В, С, D 'A, B, C, D
30 LETO=(A+B)/2: LETP=(B-A)/2: LETQ=(C+D)/2
40 LETR=(D-C)/2: LETK=SGR(1/3)
50 LETX=O+P*K: LETY=Q+R*K: GOSUB 100
60 LETX=O-P*K: GOSUB 100
70 LETY=Q-R*K: GOSUB 100
80 LETX=O+P*K: GOSUB 100
90 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I='S*R*P: GOTO 20
100 LETF=1/X/Y: LETS=S+F: RETURN: END

```

Программа 4.48.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ДВОЙНОГО ИНТЕГРАЛА ПО СЛОЖНОЙ'
15 PRINT '    ФОРМУЛЕ ГАУССА': LETS=0
20 INPUT 'ВВЕДИТЕ ПРЕДЕЛЫ А, В, С, D 'A, B, C, D
25 INPUT 'ВВЕДИТЕ ЧИСЛА М, N 'M, N
30 LETH=(B-A)/M: LETL=(D-C)/N: LETK=SGR(1/3)
40 FOR I=1 TO N: LETD=C+L
50 LETQ=(C+D)/2: LETR=L/2: LETA=A
60 FOR J=1 TO M: LETB=A+H: LETO=(A+B)/2: LETP=H/2
70 LETX=O+P*K: LETY=Q+R*K: GOSUB 140
80 LETX=O-P*K: GOSUB 140
90 LETY=Q-R*K: GOSUB 140
100 LETX=O+P*K: GOSUB 140
110 LETA=B: NEXT J
120 LETC=D: NEXT I
130 PRINT 'ЗНАЧЕНИЕ ИНТЕГРАЛА I='S*H*L/4: STOP
140 LETF=1/X/Y: LETS=S+F: RETURN: END

```

Программа 4.47.

Пример. Для интеграла (4.31) с областью интегрирования в виде одного квадрата получим $I=0,02500532089$, а с областью интегрирования в виде четырех прямоугольников — $I=0,02500594169$. Время вычисления для одного квадрата области интегрирования — около 1,5 с.

Сложная кубатурная формула Гаусса реализуется разбивкой отрезков $[a, b]$ на M и $[c, d]$ на N частей с применением для каждого элементарного прямоугольника формулы (4.32).

Программа 4.48.

Пример. Для интеграла (4.31), задав $M=N=4$ (всего 16 прямоугольников), получим $I=0,02500598247$ при $t_c=24$ с.

Аналогичным образом могут быть составлены программы для вычисления интегралов с большей кратностью. Формулы численного интегрирования для них приведены в [18, 36]. Интегралы большой кратности вычисляются методом Монте-Карло (см. [4]).

§ 4.10. Решение систем дифференциальных уравнений

Задача Коши заключается в решении систем обыкновенных дифференциальных уравнений первого порядка, представляемых в виде

$$\frac{dy_1}{dx} = F_1(x, y_1, \dots, y_j, \dots, y_N),$$

$$\frac{dy_j}{dx} = F_j(x, y_1, \dots, y_j, \dots, y_N), \quad (4.33)$$

$$\frac{dy_N}{dx} = F_N(x, y_1, \dots, y_j, \dots, y_N),$$

где $j = 1 \div N$ — номер каждой зависимой переменной y_j , x — независимая переменная. Если задача Коши решается для анализа поведения системы или объекта во времени, то x является временем ($x=t$). Решение системы (4.33) при заданных начальных условиях $x=x_0$, $y_1(x_0)=y_{10}$, $y_2(x_0)=y_{20}$, ..., $y_N(x_0)=y_{N0}$ сводится к нахождению зависимостей (интегральных кривых) $y_1(x)$, $y_2(x)$, ..., $y_j(x)$, ..., $y_N(x)$, проходящих через точки, заданные начальными условиями (x_0, y_{10}) , (x_0, y_{20}) , ..., (x_0, y_{j0}) , ..., (x_0, y_{N0}) . Задача Коши сводится к интегрированию дифференциальных уравнений. Порядок метода численного интегрирования при этом определяет и порядок метода решения (4.33).

Обобщенная форма записи каждого из уравнений системы (4.33) может быть представлена в общем виде

$$\frac{dY_j}{dx} = F_j(x, Y_j),$$

где Y_j в правой части уравнения — векторы переменных $y_1, y_2, \dots, y_j, \dots, y_N$, а F_j — правая часть каждого из уравнений (4.33). В частности, одно дифференциальное уравнение ($y = Y_j = Y_1$ и $F_j = F = F_1$) записывается в виде

$$\frac{dy}{dx} = F_1(x, y).$$

Дифференциальные уравнения высшего порядка

$$y^{(n)} = F(x, y, y', y'', \dots, y^{(n-1)}), \quad (4.35)$$

где (n) — порядок уравнения, могут быть сведены к системам вида (4.33) или (4.34) с помощью следующих преобразований:

$$\frac{dy}{dx} = y_1,$$

$$\frac{dy_1}{dx} = y_2,$$

$$\frac{dy_{n-2}}{dx} = y_{n-1}, \quad (4.36)$$

$$\frac{dy_{n-1}}{dx} = F(x, y, y_1, \dots, y_{n-1}).$$

Следовательно, решение (4.35) сводится к решению системы дифференциальных уравнений первого порядка (4.36).

Метод Эйлера — Коши — простейший метод первого порядка для численного интегрирования дифференциальных уравнений. Он реализуется следующей рекуррентной формулой:

$$Y_{j(i+1)} = Y_{ji} + hF_j(x_i, Y_{ji}),$$

где h — шаг интегрирования (приращение переменной x). Этот метод обладает большой погрешностью и имеет систематическое накопление ошибок. Погрешность метода $R \sim (h^2)$, т. е. пропорциональна h^2 .

Метод Эйлера — Коши с итерациями заключается в вычислении на каждом шаге начального значения

$$Y_{j(i+1)}^{(0)} = Y_{ji} + hF_j(x_i, Y_{ji}).$$

Затем с помощью итерационной формулы

$$Y_{j(i+1)}^{(k)} = Y_{ji} + \frac{h}{2} [F_j(x_i, Y_{ji}^{(k)}) + F_j(x_{i+1}, Y_{j(i+1)}^{(k-1)})]$$

решение уточняется. Итерации проводят до тех пор, пока не совпадает заданное число цифр результата на двух последних шагах итераций. Погрешность метода $R \sim (h^3)$. Обычное число итераций не должно превышать 3—4, иначе нужно уменьшить шаг h .

Модифицированный метод Эйлера второго порядка реализуется следующими рекуррентными формулами:

$$Y_{j(i+1)} = Y_{ji} + hF_j(x_i + h/2, Y_{j(i+1/2)}^*),$$

где $Y_{j(i+1/2)}^* = Y_{ji} + hF_j(x_i, Y_{ji})/2$. Метод дает погрешность $R \sim (h^3)$ и имеет меньшее время вычислений, поскольку вместо нескольких итераций производится вычисление только одного значения $Y_{j(i+1/2)}^*$.

Метод трапеций — одна из модификаций метода Эйлера второго порядка. Он реализуется применением на каждом шаге формулы

$$Y_{j(i+1)} = Y_{ji} + \frac{1}{2} (K_{j1} + K_{j2}),$$

где $K_{j1} = hF_j(x_i, Y_{ji})$, $K_{j2} = hF_j(x_i + h, Y_{ji} + K_{j1})$, и дает погрешность $R \sim (h^3)$. Этот метод относится к общим методам Рунге — Кутты (см. далее).

Программа 4.49.

Пр и м е р. Вычисление функций $F_j(x, Y_j)$ оформляется подпрограммой, записываемой со строки 200 и завершаемой операторами RETURN и END. При этом форма записи каждой функции следующая:

$$F(J) = \langle \text{функция } X, Y(1), Y(2), \dots, Y(N) \rangle,$$

где J — номер, соответствующий номеру j в обобщенной форме записи $F_j(x, y_j)$. Например, для решения системы дифференциальных уравнений

$$\frac{dy_1}{dx} = F_1(x, y_1, y_2) = y_2,$$

$$\frac{dy_2}{dx} = F_2(x, y_1, y_2) = \left(\frac{y_1}{x} - y_2\right) \frac{1}{x} - y_1$$

```

10 PRINT 'РЕШЕНИЕ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ '
20 PRINT ' УРАВНЕНИЙ МЕТОДОМ ТРАПЕЦИЙ'
30 INPUT 'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N=' N
40 DIM Y(N), W(N), K(N), F(N)
100 INPUT 'ВВЕДИТЕ НАЧАЛЬНОЕ X0=' X
110 FOR J=1 TO N:PRINT!2.0!'ВВЕДИТЕ НАЧАЛЬНОЕ Y0('J')'
120 INPUT W(J):LETY(J)=W(J):NEXT J
130 INPUT 'ВВЕДИТЕ ШАГ H=' H
140 GOSUB 200:FOR J=1 TO N
150 LETK(J)=H*F(J):LETY(J)=W(J)+K(J):NEXT J
160 LETX=X+H:PRINT!F1.9!'ДЛЯ X='X:GOSUB 200
170 FOR J=1 TO N:LETY(J)=W(J)+(K(J)+H*F(J))/2
180 PRINT!2.0!'Y('J')='!F1.9!Y(J)
190 LETW(J)=Y(J):NEXT J:GOTO 140
200 LETF(1)=Y(2):LETF(2)=(Y(1)/X-Y(2))/X-Y(1)
210 RETURN:END
    
```

подпрограмму следует записать в виде

$$F(1) = Y(2),$$

$$F(2) = (Y(1)/X - Y(2))/X - Y(1)$$

(см. также выражения в строке 200 программы 4.49). Для $N=2$; $h=0,1$; $x_0=0,2$; $y_{10}=0,099500833$ и $y_{20}=0,49235$ получим:

x	y_1	y_2
0,3	0,148367183	0,4830282264
0,4	0,1961203205	0,4701529365
0,5	0,2424068609	0,4537779586
...		
1,0	0,440372468	0,3249711437

и т. д.

Время счета на каждом шаге около 1,5 с. Не рекомендуется применять этот метод, если возможны изменения $y_j(x)$ с сильно различающейся крутизной (например, одни

$$K_{3j} = hF_j \left(x_i + \frac{h}{2}, Y_{ii} + \frac{1}{2} K_{2j} \right);$$

$$K_{4j} = hF_j(x_i + h, Y_{ii} + K_{3j});$$

$$Y_{i(i+1)} = Y_{ii} + \frac{1}{6} (K_{1j} + 2K_{2j} + 2K_{3j} + K_{4j}).$$

При переходе от одной формулы к другой задаются или вычисляются соответствующие значения x и Y_j и находятся по подпрограмме значения функций $F_j(x, Y_j)$.

Решение одного дифференциального уравнения методом Рунге — Кутта производится по приведенным формулам, если в них опустить индекс j , а из алгоритма исключить циклы. Последнее резко упрощает программу и позволяет получить минимально возможное время счета.

Программа 4.50.

```

10 PRINT 'РЕШЕНИЕ ОДНОГО ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ '
20 PRINT ' МЕТОДОМ РУНГЕ-КУТТА'
30 INPUT 'ЗАДАЙТЕ ШАГ H=' H:INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ X0=' X
40 INPUT 'ВВЕДИТЕ НАЧАЛЬНОЕ Y0=' Z:LETY=Z
50 GOSUB 100:LETA=F*H:LETX=X+H/2:LETY=Z+A/2
60 GOSUB 100:LETB=F*H:LETY=Z+B/2
70 GOSUB 100:LETC=F*H:LETX=X+H/2:LETY=Z+C
80 GOSUB 100:LETY=Z+(A+2*(B+C)+H*F)/6:LETT=Y
90 PRINT 'ДЛЯ X=' X:PRINT 'Y=' Y:GOTO 50
100 LETF=-Y:RETURN:END
    
```

участки кривых $y_j(x)$ плавные, а другие резкие), поскольку, при этом возможно возникновение численной неустойчивости решения. Ее устранение может потребовать выбора очень малого шага h , что значительно увеличивает общее время вычислений.

Метод Рунге — Кутта четвертого порядка является наиболее распространенным методом решения систем (4.34) при шаге $h = \text{const}$. Его достоинством является высокая точность — погрешность $R \sim (h^5)$ — и меньшая склонность к возникновению неустойчивости решения. Алгоритм реализации метода Рунге — Кутта заключается в циклических вычислениях $Y_{i(i+1)}$ на каждом $i+1$ шаге по следующим формулам:

$$K_{1j} = hF_j(x_i, Y_{ii}); K_{2j} = hF_j \left(x_i + \frac{h}{2}, Y_{ii} + \frac{1}{2} K_{1j} \right);$$

Пример. Для дифференциального уравнения $y' = -y/\tau = -y$ при $\tau=1$, $h=0,1$, $x_0=0$ и $y_0=1$ будем получать $y(0,1) = 0,9048375$, $y(0,2) = 0,8187309014$, $y(0,3) = 0,740818422$, $y(0,4) = 0,6703202889$, $y(0,5) = 0,6065309344$. Время выдачи одного значения $y(x)$ около 1 с. Подпрограмма вычисления

$$F = dy/dx$$

записывается со строки 100.

Реализация решения системы из N дифференциальных уравнений иллюстрируется программой 4.51. Максимальное число N зависит от сложности уравнений и объема ОЗУ ЭВМ. Для ПЭВМ типовые максимальные значения N лежат от 10—20 (для ПЭВМ класса Pocket Computers) до 50—100 (для профессиональных ЭВМ).

```

10 PRINT'РЕШЕНИЕ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ'
20 PRINT'УРАВНЕНИЙ МЕТОДОМ РУНГЕ-КУТТА'
30 INPUT'ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N='N
40 DIM Y(N),A(N),K(N),F(N),W(N)
50 INPUT'ЗАДАЙТЕ ШАГ H='H
60 INPUT'ЗАДАЙТЕ НАЧАЛЬНОЕ X0='X
70 FOR J=1 TO N:PRINT!2.0!'ЗАДАЙТЕ НАЧАЛЬНОЕ Y0('J')
80 INPUT W(J):LETY(J)=W(J):NEXT J
90 GOSUB 200:FOR J=1 TO N:LETU=H*F(J)
100 LETK(J)=U:LETY(J)=W(J)+U/2:NEXT J
110 LETX=X+H/2:GOSUB 200:FOR J=1 TO N
120 LETU=H*F(J):LETK(J)=K(J)+2*U
130 LETY(J)=W(J)+U/2:NEXT J
140 GOSUB 200:FOR J=1 TO N:LETU=H*F(J)
150 LETK(J)=K(J)+2*U:LETY(J)=W(J)+U:NEXT J
160 LETX=X+H/2:PRINT!F1.9!'ДЛЯ X='X:GOSUB 200
170 FOR J=1 TO N:LETY(J)=W(J)+(K(J)+H*F(J))/6
180 PRINT!2.0!'Y('J')='!F1.9!Y(J)
190 LETW(J)=Y(J):NEXT J:GOTO 90
200 LETF(1)=Y(2):LETF(2)=(Y(1)/X-Y(2))/X-Y(1)
210 RETURN:END
    
```

Пример. Для системы уравнений (см. подпрограмму, записанную со строки 200)

$$F_1 = \frac{dy_1}{dx} = y_2;$$

$$F_2 = \frac{dy_2}{dx} = \left(\frac{y_1}{x} - y_2\right) \frac{1}{x} - y_1$$

при $N=2$, $h=0,1$, $x_0=0,2$, $y_{10}=0,099500833$ и $y_{20}=0,49235$ будем получать следующие значения x , y_1 и y_2 :

x	y_1	y_2
0,3	0,1483051278	0,4831085139
0,4	0,1960019791	0,4702289086
0,5	0,2422342383	0,453843868
...
1	0,4399784882	0,3250861127
...
1,5	0,5578420795	0,1398423554
...
2,0	0,5766255981	-0,06446334779

Ввиду особого значения и широкого применения дифференциальных уравнений второго порядка полезно иметь специальную программу для их решения.

Метод Рунге — Кутта для дифференциального уравнения второго порядка

```

10 PRINT'РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ ВТОРОГО'
20 PRINT'ПОРЯДКА МЕТОДОМ РУНГЕ-КУТТА'
30 INPUT'ВВЕДИТЕ ШАГ H='H:INPUT'ВВЕДИТЕ НАЧАЛЬНОЕ X0='X
40 INPUT'ВВЕДИТЕ НАЧАЛЬНОЕ Y0='Y
50 INPUT'ВВЕДИТЕ НАЧАЛЬНОЕ DY/DX0='U:LETY=W:LETZ=W
60 GOSUB 150:LETA=H*F:LETX=X+H/2
70 LETY=W+U*H/2+A*H/8:LETZ=U+A/2
80 GOSUB 150:LETB=H*F:LETZ=U+B/2
90 GOSUB 150:LETC=H*F:LETX=X+H/2
100 LETY=W+H*(U+H*C/2):LETZ=U+C:GOSUB 150
110 LETY=W+H*(U+(A+B+C)/6):LETW=Y
120 LETZ=U+(A+(B+C)*2+H*F)/6:LETU=Z
130 PRINT'ДЛЯ X='X:PRINT'Y='Y
140 PRINT'DY/DX='Z:GOTO 60
150 LETF=-Y+(1-Y*Y)*Z*20:RETURN:END
    
```

вида

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

имеющий погрешность $R \sim (h^5)$, реализуется с помощью следующих формул [36]:

$$K_1 = hF(x_i; y_i; y'_i);$$

$$K_2 = hF\left(x_i + \frac{h}{2}; y_i + \frac{h}{2}y'_i + \frac{h}{8}K_1; y'_i + \frac{K_1}{2}\right);$$

$$K_3 = hF\left(x_i + \frac{h}{2}; y_i + \frac{h}{2}y'_i + \frac{h}{8}K_1; y'_i + \frac{K_2}{2}\right);$$

$$K_4 = hF\left(x_i + h; y_i + hy'_i + \frac{h}{2}K_3; y'_i + K_3\right);$$

$$y_{i+1} = y_i + h\left[y'_i + \frac{1}{6}(K_1 + K_2 + K_3)\right];$$

$$y'_{i+1} = y'_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4).$$

Перед началом вычислений надо задать шаг h и начальные значения x_0 , $y(x_0) = y_0$ и $y'(x_0) = y'_0$.

Программа 4.52.

Пр и м е р. Д ля дифференциального уравнения (см. подпрограмму, записанную в строке 150)

$$F = y'' = -y + (1 - y^2) (20y')$$

при $h = 0,003125$, $x_0 = 0$, $y_0 = 2,0077$ и $y'_0 = 0$ будем получать:

x	y	y'
0,003125	2,007690786	-0,00571555801
0,00625	2,007665302	-0,01044479113
0,009375	2,007626354	-0,01435793978
0,0125	2,00757266	-0,01759589125
0,015625	2,007516959	-0,02027523125

Автоматическое изменение шага в ходе решения систем дифференциальных уравнений необходимо, если решение требуется получить с заданной точностью. При высокой точности (погрешность $\epsilon \approx E = 1 \cdot 10^{-3}$) и решении в виде кривых с сильно различающейся крутизной автоматическое изменение шага обеспечивает уменьшение общего числа шагов в несколько раз, резко уменьшает вероятность возникновения числовой неустойчивости, дает более равномерное расположение точек графика кривых (решений) при их выводе на печать.

Метод Рунге — Кутта с автоматическим изменением шага заключается в том, что после вычисления $Y_{j(i+1)}$ с шагом h все вычисления проводятся повторно с шагом $h/2$. Полученный результат $Y_{j(i+1)}$ сравнивается с $Y_{j(i+1)}$. Если $|Y_{j(i+1)} - Y_{j(i+1)}^*| < \epsilon$, вычисления продолжают с шагом h , в противном случае шаг уменьшают. Если это неравенство слишком сильное, шаг, напротив, увеличивают. При той же погрешности $R \sim (h^5)$ лучшие результаты дает описанный ниже метод.

Метод Рунге — Кутта — Мерсона с автоматическим изменением шага обеспечивает приближенную оценку погрешности на каждом шаге интегрирования. Погрешность интегрирования имеет порядок h^5 [31]. Этот метод реализуется следующим алгоритмом.

1. Задаем число уравнений N , погрешность $\epsilon = E$, начальный шаг интегрирования $h = H$ и начальное значение $x_0, y_{10}, \dots, y_{N0}$.

2. С помощью 5 циклов с управляющей переменной $J = 1, 2, \dots, N$ вычисляем коэффициенты:

$$K_{0j} = hF_j(x_j; Y_{j0});$$

$$K_{1j} = hF_j\left(x_j + \frac{1}{3}h; Y_{j0} + \frac{1}{3}K_{0j}\right);$$

$$K_{2j} = hF_j\left(x_j + \frac{1}{3}h; Y_{j0} + \frac{1}{6}K_{0j} + \frac{1}{6}K_{1j}\right);$$

$$K_{3j} = hF_j\left(x_j + \frac{1}{2}h; Y_{j0} + \frac{1}{8}K_{0j} + \frac{3}{8}K_{2j}\right);$$

$$K_{4j} = hF_j\left(x_j + h; Y_{j0} + \frac{1}{2}K_{0j} - \frac{3}{2}K_{2j} + 2K_{3j}\right).$$

3. Находим (в последнем цикле) значение

$$Y_{j(i+1)} = Y_{ji} + (K_{0j} + 4K_{2j} + K_{4j})/6$$

и погрешность

$$R_{j(i+1)} = (-2K_{0j} + 9K_{2j} - 8K_{3j} + K_{4j})/30.$$

4. Проверяем выполнение условий

$$|R_{j(i+1)}| \leq E, |R_{j(i+1)}| \geq E/30.$$

Если первое условие не выполняется, делим шаг h на 2 и повторяем вычисления с п. 2, восстановив начальные значения Y_{ji} . Если это условие выполняется и выполняется второе условие, значения $x_{i+1} = x_i + h$ и $Y_{j(i+1)}$ выводятся на печать. Если второе условие не выполняется, шаг h увеличивается вдвое и вычисления опять повторяются с п. 2.

Таким образом, $Y_{j(n+1)}$ выводится на печать только при одновременном выполнении условий этого пункта.

Программа 4.53.

Пр и м е р. Для системы дифференциальных уравнений

$$F_1 = \frac{dy_1}{dx} = y_1 + y_2 - x^2 + x - 2,$$

$$F_2 = \frac{dy_2}{dx} = -2y_1 + 4y_2 + 2x^2 - 4x - 7$$

при $N = 2$, $h = 0,1$, $E = 1 \cdot 10^{-7}$, $x_0 = 0$, $y_{10} = 0$ и $y_{20} = 2$ получим:

x	y_1	y_2
0,025	$6,25 \cdot 10^{-4}$	2,025
0,05	$2,5 \cdot 10^{-3}$	2,05
0,075	$5,625 \cdot 10^{-3}$	2,075
0,1	$1 \cdot 10^{-2}$	2,1

Следовательно, в данном случае имеет место автоматическое уменьшение шага. Для $h = 0,02$ и $E = 1 \cdot 10^{-3}$ получим:

x	y_1	y_2
0,02	$6,731133879 \cdot 10^{-4}$	2,020963777
0,04	$1,898781104 \cdot 10^{-3}$	2,041032153
0,08	$6,756088366 \cdot 10^{-3}$	2,081182964
0,16	$2,609871963 \cdot 10^{-2}$	2,161549932
0,32	$1,033398344 \cdot 10^{-1}$	2,322641762
0,64	$4,127175414 \cdot 10^{-1}$	2,74775972
0,96	$9,309436052 \cdot 10^{-1}$	2,981657806
1,28	1,665020406	3,338953272

В данном случае шаг автоматически увеличивается от значения $h = 0,02$ до $h = 0,32$. Время вычислений на одном шаге (кроме начального при уменьшении h) составляет около 5 с.

Подпрограмма вычисления производных записывается со строки 400.

Как отмечалось, погрешность $R_{j(i+1)}$ на каждом шаге метода Рунге — Кутта — Мерсона оценивается приближенно. При решении нелинейных дифференциальных уравнений истинная погрешность может отличаться в несколько раз от заданной E .

```

05 PRINT 'РЕШЕНИЕ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ'
10 PRINT 'МЕТОДОМ РУНГЕ-КУТТА-МЕРСОНА С АВТОМАТИЧЕСКИМ'
15 PRINT 'ВЫБОРОМ ШАГА': PRINT 12.0!
40 INPUT 'ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N=': N
45 DIM Y(N), W(N), A(N), C(N), D(N), E(N), F(N)
50 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕНИЙ E=': E1
60 INPUT 'ЗАДАЙТЕ НАЧАЛЬНЫЙ ШАГ H=': H
70 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ X0=': X
80 FOR J=1 TO N: PRINT 'ВВЕДИТЕ НАЧАЛЬНОЕ Y0('J)=':
90 INPUT W(J): LET Y(J)=W(J): NEXT J
110 LET E3=0: GOSUB 400: LET D=0: FOR J=1 TO N
120 LET A(J)=F(J)*H: LET Y(J)=W(J)+A(J)/3: NEXT J
130 LET X=X+H/3: GOSUB 400: FOR J=1 TO N
140 LET Y(J)=W(J)+(A(J)+F(J)*H)/6: NEXT J
150 GOSUB 400: FOR J=1 TO N: LET C(J)=F(J)*H
160 LET Y(J)=W(J)+A(J)/8+.375*C(J): NEXT J
170 LET X=X+H/6: GOSUB 400: FOR J=1 TO N
180 LET D(J)=F(J)*H: LET Y(J)=W(J)+A(J)/2-1.5*C(J)+2*D(J)
190 NEXT J: LET X=X+H/2: GOSUB 400
200 FOR J=1 TO N: LET E(J)=F(J)*H
210 LET Y(J)=W(J)+(A(J)+4*D(J)+E(J))/6
220 LET E2=ABS(-2*A(J)+9*C(J)-8*D(J)+E(J))/30
230 IF E2<=E1 THEN 250
240 LET E3=1: GOTO 260
250 IF E2<E1/20 THEN LET D=D+1
260 NEXT J: IF E3=0 THEN 290
270 LET X=X+H: FOR J=1 TO N: LET Y(J)=W(J): NEXT J
280 LET H=H/2: GOTO 110
290 IF D=N THEN LET H=H+H
300 PRINT 'F1.9! ДЛЯ X=': X: FOR J=1 TO N
310 PRINT '2.0! Y('J)=': Y(J): F1.9! Y(J)
320 LET W(J)=Y(J): NEXT J: GOTO 110
400 LET F(1)=Y(1)+Y(2)-X*X+X-2
410 LET F(2)=-2*Y(1)+4*Y(2)+2*X*X-4*X-7
420 RETURN: END

```

Метод Рунге — Кутта — Фельберга с автоматическим изменением шага дает более точную оценку погрешности на каждом шаге и реализуется последовательным циклическим вычислением по следующим формулам [31]:

$$K_{0j} = hF_j(x_j, Y_{j\bar{u}});$$

$$K_{1j} = hF_j\left(x_j + \frac{2}{9}h; Y_{j\bar{u}} + \frac{2}{9}K_{0j}\right);$$

$$K_{2j} = hF_j\left(x_j + \frac{1}{3}h; Y_{j\bar{u}} + \frac{1}{12}K_{0j} + \frac{1}{4}K_{1j}\right);$$

$$K_{3j} = hF_j\left(x_j + \frac{3}{4}h; Y_{j\bar{u}} + \frac{69}{128}K_{0j} + \frac{143}{128}K_{1j} + \frac{135}{64}K_{2j}\right);$$

$$K_{4j} = hF_j\left(x_j + h; Y_{j\bar{u}} + \frac{17}{12}K_{0j} + \frac{27}{4}K_{1j} - \frac{27}{5}K_{2j} + \frac{16}{15}K_{3j}\right);$$

$$K_{5j} = hF_j\left(x_j + \frac{5}{6}h; Y_{j\bar{u}} + \frac{65}{432}K_{0j} - \frac{5}{16}K_{1j} + \frac{13}{16}K_{2j} + \frac{4}{27}K_{3j} + \frac{5}{144}K_{4j}\right);$$

$$Y_{j(u+1)} = Y_{j\bar{u}} + \frac{1}{9}K_{0j} + \frac{9}{20}K_{2j} + \frac{16}{45}K_{3j} + \frac{1}{12}K_{4j}. \quad (4.37)$$

Погрешность

$$R_{j(u+1)} = \frac{1}{150}K_{0j} - \frac{3}{100}K_{2j} + \frac{16}{75}K_{3j} + \frac{1}{20}K_{4j} - \frac{6}{25}K_{5j}$$

в этом методе — разность приращений $Y_{j(u+1)}$, вычисленных по двум формулам: порядка $n=4$ (4.37) и порядка $n+1$. Последняя формула не приводится, но использована для вычисления $R_{j(u+1)}$. Если $R_{j(u+1)} > E$, шаг h уменьшается вдвое, если $R_{j(u+1)} < E/20$, он увеличивается вдвое. Этот метод имеет четвертый порядок.


```

05 PRINT 'РЕШЕНИЕ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ'
10 PRINT 'МЕТОДОМ РУНГЕ-КУТТА-ФЕЛЬБЕРГА С АВТОМАТИЧЕСКИМ'
15 PRINT 'ВЫБОРОМ ШАГА': PRINT!2,0!
40 INPUT 'ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N=' N
45 DIM Y(N), L(N), W(N), A(N), B(N), C(N), D(N), E(N), F(N)
50 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕННЙ E=' E1
60 INPUT 'ЗАДАЙТЕ НАЧАЛЬНЫЙ ШАГ H=' H
70 INPUT 'ЗАДАЙТЕ НАЧАЛЬНОЕ X0=' X
80 FOR J=1 TO N:PRINT 'ВВЕДИТЕ НАЧАЛЬНОЕ Y0('J')='
90 INPUT W(J):LETY(J)=W(J):NEXT J
110 LETE3=0:GOSUB 400:LETD=0:FOR J=1 TO N
120 LETA(J)=F(J)*H:LETY(J)=W(J)+2*A(J)/9:NEXT J
130 LETX=X+2*H/9:GOSUB 400:FOR J=1 TO N
140 LETB(J)=H*F(J):LETY(J)=W(J)+A(J)/12+B(J)/4:NEXT J
150 LETX=X+H/9:GOSUB 400:FOR J=1 TO N
160 LETC(J)=H*F(J):LETY(J)=W(J)+(34.5*A(J)-121.5*B(J)+135*C(J))/64
170 NEXT J:LETX=X+H/2.4:GOSUB 400:FOR J=1 TO N
180 LETD(J)=H*F(J):LETO=W(J)-17*A(J)/12+6.75*B(J)
185 LETY(J)=O-5.4*C(J)+16*D(J)/15:NEXT J
190 LETX=X+H/4:GOSUB 400:FOR J=1 TO N
195 LETE(J)=H*F(J):LETO=W(J)+65*A(J)/432-.3125*B(J)
200 LETY(J)=O+.8125*C(J)+4*D(J)/27+5*E(J)/144:NEXT J
205 LETX=X-H/6:GOSUB 400:FOR J=1 TO N
210 LETL(J)=H*F(J):LETY(J)=W(J)+A(J)/9+.45*C(J)+16*D(J)/45+E(J)/12
220 LETO=A(J)/150-.03*C(J)+16*D(J)/75+E(J)/20
230 LETE2=ABS(O-.24*L(J)):IF E2<=E1 THEN 250
240 LETE3=1:GOTO 260
250 IF E2<E1/20 THEN LETD=D+1
260 NEXT J:LETX=X+H/6:GOSUB 400:IF E3=0 THEN 290
270 LETX=X-H:FOR J=1 TO N:LETY(J)=W(J):NEXT J
280 LETH=H/2:GOTO 110
290 IF D=N THEN LETH=H+H
300 PRINT!F1.9!' ДЛЯ X='X:FOR J=1 TO N
310 PRINT!2,0!'Y('J')='!F1.9!Y(J)
320 LETW(J)=Y(J):NEXT J:GOTO 110
400 LETF(1)=Y(1)+Y(2)-X*X+X-2
410 LETF(2)=-2*Y(1)+4*Y(2)+2*X*X-4*X-7
420 RETURN:END

```

Пример. Для системы дифференциальных уравнений и данных к программе 4.53 будем получать:

x	y ₁	y ₂
0,05	2,499985857 · 10 ⁻³	2,049999950
0,1	9,999968178 · 10 ⁻³	2,099999890
0,15	2,249994626 · 10 ⁻²	2,149999819
0,2	3,999991927 · 10 ⁻²	2,199999736
0,25	6,249988623 · 10 ⁻²	2,249999637

Время счета для одной точки около 10 с, т. е. вдвое больше, чем у программы 4.53. Однако вдвое больше и выбранный программой шаг h. В целом скорость счета у программ 4.53 и 4.54 примерно одинакова и зависит от характера решаемой системы дифференциальных уравнений.

Одношаговые методы решения систем дифференциальных уравнений, к которым относятся все описанные выше методы, основаны на вычислениях по рекуррентным формулам, содержащим данные, полученные из решения на одном предшествующем шаге. Эти методы обеспечивают автоматическое начало вычислений при заданных начальных условиях и изменение (в том числе автоматическое) шага в ходе вычислений.

Многошаговые методы решения дифференциальных уравнений базируются на использовании данных решения на нескольких предшествующих шагах. Это позволяет по-

высвить скорость вычислений. Однако для начала вычислений приходится выполнять одношаговыми методами несколько первых шагов. Аналогично это делается при каждой смене шага интегрирования.

Методы прогноза и коррекции — разновидности многошаговых методов, при которых решение вначале проводится по формуле прогноза, а затем уточняется по формуле коррекции.

Ввиду сложности программной реализации многошаговых методов, а также неявных методов численного интегрирования (последние лишены численной неустойчивости решения), они очень редко используются при решении задач на персональных ЭВМ. Алгоритмы их реализации описаны в [10, 18, 24, 36].

§ 4.11. Гармонический синтез

Гармонический синтез — получение колебаний сложной формы путем суммирования их гармонических составляющих (гармоник). Гармонический синтез может иметь как самостоятельное значение, так и использоваться как средство тригонометрической интерполяции функций (совместно со спектральным анализом — см. § 5.1).

Гармонический синтез с помощью ряда Фурье (1 вариант) заключается в вычислении ограниченного m гармониками ряда Фурье

$$y(t) = \frac{A_0}{2} + \sum_{k=1}^m (A_k \sin 2\pi k f_1 t + B_k \cos 2\pi k f_1 t),$$

где t — время, f_1 — частота первой гармоники, A_k и B_k — коэффициенты ряда Фурье.

Программа 4.55.

```

10 PRINT 'ГАРМОНИЧЕСКИЙ СИНТЕЗ 1'
20 INPUT 'ВВЕДИТЕ ЧИСЛО ГАРМОНИК M=':M:DIM A(M),B(M)
30 PRINT 'ВВЕДИТЕ A 0':INPUT A
40 FOR N=1 TO M:PRINT!2.0!'ВВЕДИТЕ A'N
50 INPUT A(N):PRINT 'ВВЕДИТЕ B'N':INPUT B(N):NEXT N
60 INPUT 'ВВЕДИТЕ ЧАСТОТУ F=':F
70 INPUT 'ВВЕДИТЕ ВРЕМЯ T=':T:LETC=2**PI*F*T:LETS=0
80 FOR N=1 TO M: LETD=C*N
90 LETS=S+(A(N)*COS(D)+B(N)*SIN(D)):NEXT N
100 LETY=A/2+S:PRINT 'Y(T)='!F1.9!Y:GOTO 70:END

```

Пример. Для $A_0=4$, $A_1=10$, $B_1=5$, $A_2=8$, $B_2=2$, $A_3=4$, $B_3=1$, $A_4=1$, $B_4=0,2$, $A_5=0,1$, $B_5=0,05$ и $f_1=1$ будем получать: $y(0)=25,1$; $y(0,1)=16,32687379$; $y(0,2)=-0,9348395361$.

Гармонический синтез с помощью ряда Фурье (2 вариант) сводится к вычислению ряда

$$y(t) = \frac{A_0}{2} + \sum_{k=1}^m A_k \cos(2\pi k f_1 t + Q_k),$$

заданного гармониками с амплитудой A_k и фазовым углом Q_k (рад).

Программа 4.56.

```

10 PRINT 'ГАРМОНИЧЕСКИЙ СИНТЕЗ 2'
20 INPUT 'ВВЕДИТЕ ЧИСЛО ГАРМОНИК M=':M:DIM A(M),Q(M)
30 PRINT 'ВВЕДИТЕ A 0':INPUT A
40 FOR N=1 TO M:PRINT!2.0!'ВВЕДИТЕ A'N
50 INPUT A(N):PRINT!2.0!'ВВЕДИТЕ ФАЗУ В РАДИАНАХ Q'N
55 INPUT Q(N):NEXT N
60 INPUT 'ВВЕДИТЕ ЧАСТОТУ F=':F
70 INPUT 'ВВЕДИТЕ ВРЕМЯ T=':T:LETC=2**PI*F*T:LETS=0
80 FOR N=1 TO M:LETS=S+A(N)*COS(C*N+Q(N)):NEXT N
100 LETY=A/2+S:PRINT 'Y(T)='!F1.9!Y:GOTO 70:END

```

Пример. Для $A_0=4$, $A_1=10$, $Q_1=0$, $A_2=8$, $Q_2=0,5$, $A_3=5$, $Q_3=0,8$, $A_4=2$, $Q_4=2$, $A_5=0,5$, $Q_5=3$ и $f_1=1$ будем получать: $y(0,1)=4,223677837$; $y(0,2)=-2,576641635$.

Аналогичным образом осуществляется гармонический синтез при других тригонометрических рядах. Например, при ряде Котельникова с ограниченным числом членов [18]

$$y(t) = \sum_{k=0}^m y(k \Delta t) \frac{\sin \pi (t/\Delta t - k)}{\pi (t/\Delta t - k)},$$

где Δt — шаг, с которым задаются отсчеты $y(k \Delta t)$. В некоторых случаях гармонический синтез упрощается. Так, если предполагаемая функция $y(t)$ симметрична относительно оси t , то амплитуды всех четных гармоник равны нулю.

§ 4.12. Вычисление собственных значений и векторов матриц

Собственные значения λ_i ($i=1, 2, \dots, n$) квадратной матрицы $A = [a_{ij}]$ n -го порядка есть числа (действительные или комплексные), удовлетворяющие условиям $Ax = \lambda_i x$. В механике собственные значения характеризуют, например, внутренние напряжения в объекте, находящимся под воздействием

внешних усилий, действующих со всех сторон. В электро-радиотехнических устройствах собственные значения матриц определяют характеристические постоянные времени и режимы работы этих устройств. Все это говорит о практической важности задачи нахождение собственных значений [4, 24, 41].

Определитель

$$D(\lambda) = \begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nn} - \lambda \end{vmatrix}$$

называется *характеристическим многочленом*

(*полиномом*) матрицы A . Уравнение $Ax = \lambda x$ эквивалентно уравнению $(A - \lambda E)x = 0$. Уравнение $\det(A - \lambda E) = 0$ называется *характеристическим многочленом матрицы A* :

$$\det(A - \lambda E) = P(\lambda) = -a_1 - a_2 \lambda - a_3 \lambda^2 - \dots - a_n \lambda^{n-1} + \lambda^n = 0.$$

Его корни определяют все собственные значения λ_i .

Каждому λ_i соответствует принадлежащий ему *собственный вектор* Y_i с компонентами $y_{i1}, y_{i2}, \dots, y_{in}$. Они удовлетворяют решению системы уравнений $A Y_i = \lambda_i Y_i$.

Коэффициенты a_i характеристического полинома $P(\lambda)$ находятся *методом Данилевского*. Для этого рассматривается специальным образом преобразованная последовательность матриц $A^{(1)}, A^{(2)}, \dots, A^{(n)}$, при-

чем преобразование (см. ниже) направлено на то, чтобы получить матрицу $A^{(N)}$, у которой коэффициенты a_i характеристического полинома оказываются ее последним столбцом. Преобразования $A^{(k)}$ в $A^{(k+1)}$ выполняются по формулам

$$A_{ij}^{(k+1)} = B_{ij}^{(k)}, \quad j \neq k+1, \quad A_{i, k+1}^{(k+1)} = \sum_{j=1}^N B_{ij}^{(k)} A_{ij}^{(k)},$$

где $B_{k+1, i}^{(k+1)} = A_{k+1, i}^{(k)} / A_{k+1, k}^{(k)}$ и $B_{ij}^{(k)} = A_{ij}^{(k)} - A_{ik}^{(k)} B_{k+1, i}^{(k)}$ ($i \neq k+1$), причем $i=1, 2, \dots, N$; $j=1, 2, \dots, N$. Для экономии памяти ПЭВМ вспомогательная матрица B заносится на место матрицы A .

Программа 4.57.

```

05 PRINT 'ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТОВ ХАРАКТЕРИСТИЧЕСКОГО ПОЛИНОМА'
10 PRINT 'МАТРИЦЫ МЕТОДОМ ДАНИЛЕВСКОГО'
20 INPUT 'ЗАДАЙТЕ ПОРЯДОК МАТРИЦЫ N=':N:DIM A(N,N)
25 FOR I=1 TO N:FOR J=1 TO N
30 PRINT!2.0!'ВВЕДИТЕ A(I,J)':INPUT A(I,J):NEXT J:NEXT I:LETQ=1
40 FOR K=Q TO N-1:LETL=0:FOR I=K+1 TO N
50 IF ABS(L)-ABS(A(I,K))<0 THEN LETL=A(I,K):LETP=I
60 NEXT I:IF L<>0 THEN 90
70 FOR J=Q TO K:LETA(Q,J)=A(J,K):NEXT J
80 LETQ=K+1:PRINT!2.0!'K='K:GOTO 40
90 IF K+1=P THEN 140
100 FOR J=K TO N:LETR=A(K+1,J)
110 LETA(K+1,J)=A(P,J):LETA(P,J)=R:NEXT J
120 FOR J=Q TO N:LETR=A(J,K+1)
130 LETA(J,K+1)=A(J,P):LETA(J,P)=R:NEXT J
140 FOR J=Q TO N:LETA(Q,J)=A(J,K):NEXT J
150 LETC=A(K+1,K):FOR J=K TO N
160 LETA(K+1,J)=A(K+1,J)/C:FOR I=Q TO N
170 IF I=K+1 THEN 190
180 LETA(I,J)=A(I,J)-A(Q,I)*A(K+1,J)
190 NEXT I:NEXT J
200 FOR I=Q TO N:LETS=0:FOR J=K+1 TO N
210 LETS=S+A(I,J)*A(Q,J):NEXT J
220 IF I-1<=K THEN 240
230 LETD=0:GOTO 260
240 IF I=Q THEN LETD=0:GOTO 260
250 LETD=A(Q,I-1)
260 LETA(I,K+1)=S+D:NEXT I:NEXT K
270 FOR J=Q TO N:LETA(Q,J)=A(J,N):NEXT J
280 FOR J=1 TO N:PRINT!2.0!'A'J!F1.9!'='A(Q,J)
290 NEXT J:END

```

Пример. Если A — невырожденная матрица, программа 3.69 дает значения $A_i = a_i$ характеристического полинома. Например, для матрицы

$$A = \begin{bmatrix} -5,509882 & 1,870086 & 0,422908 \\ 0,287865 & -11,811654 & 5,7119 \\ 0,049099 & 4,308033 & -12,970687 \\ 0,006235 & 0,269851 & 1,397369 \end{bmatrix}$$

получаем $A_1 = a_1 = -12296,55057$, $A_2 = a_2 = -5349,455515$, $A_3 = a_3 = -797,2787648$ и $A_4 = a_4 = -47,88843$.

Если A — вырожденная матрица, то массив $A_i = a_i$ содержит коэффициенты с обратными знаками всех составных полиномов. Идентификатором полиномов в этом случае являются значения переменной k , выдавае-

мые программой. Степень первого полинома равна первому значению k , степень второго — разности второго и первого значений k и т. д. (степень последнего полинома равна разности $N = n$ и последнего значения k). Так, если программа выдает $k_1 = 2$, $k_2 = 3$ ($N = 5$), а массив a_i выдается в виде пяти чисел

$$\begin{matrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ a_{21}^{(1)} & a_{22}^{(2)} & a_{33}^{(2,3)} & a_{34}^{(3)} & a_{55}^{(3)}, \end{matrix}$$

то степень первого полинома $k_1 = 2$ и, следовательно,

$$P_1(\lambda) = -a_{21}^{(1)} - a_{22}^{(1)}\lambda + \lambda^2.$$

Степень второго полинома $k_2 - k_1 = 1$ и

$$P_2(\lambda) = -a_{33}^{(2)} + \lambda.$$

Степень третьего полинома $N - k_2 = 2$ и

$$P_3(\lambda) = -a_{54}^{(3)} - a_{55}^{(3)}\lambda + \lambda^2.$$

Для матрицы

$$A = \begin{bmatrix} 1 & 1 & 4 & 3 \\ 1 & 1 & 2 & 1 \\ 0 & 0 & 4 & 3 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

программа 4.57 выдает $k = k_1 = 2$, $A_1 = 0$, $A_2 = 2$, $A_3 = 2$ и $A_4 = 5$. Составляя полиномы по описанным для этого случая правилам, получим

$$P_1(\lambda) = 0 - 2\lambda + \lambda^2 = -2\lambda + \lambda^2$$

и

$$P_2(\lambda) = -2 - 5\lambda + \lambda^2.$$

Жорни полиномов $I(\lambda)$ могут быть вычислены по программам, описанным в § 4.5.

Максимальное собственное значение матрицы (действительное) и принадлежащий ему собственный вектор могут вычисляться степенным методом по следующему алгоритму.

1. По формуле $Y^{(c)} = Ay^{(c-1)}$, где

$$Y^{(c)} = \begin{bmatrix} y_1^{(c)} \\ y_2^{(c)} \\ \dots \\ y_n^{(c)} \end{bmatrix}$$

и

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix},$$

вычисляется вектор $Y^{(c)}$ (перед началом вычислений задается произвольный исходный вектор $Y^{(0)}$).

2. Находятся приближения к наибольшему собственному значению

$$\mu_c = \frac{1}{n} \sum_{i=1}^n y_i^{(c)} / y_i^{(c-1)}.$$

3. Вектор $Y^{(c)}$ нормируется, т. е. вместо него получается вектор

$$\tilde{Y}^{(c)} = \frac{1}{\max Y_i^{(c)}} Y^{(c)}, \quad 1 \leq i \leq n.$$

4. Проверяется выполнение условия

$$|\mu_c - \mu_{c-1}| < \varepsilon.$$

где ε — заданная погрешность вычисления максимального по модулю собственного значения. Если это условие выполняется, считаем μ_c максимальным по модулю собственным значением, а вектор $\tilde{Y}^{(k)}$ — принадлежащим ему собственным вектором. Если данное условие не выполняется, идем к п. 1, приняв за $Y^{(k-1)}$ вектор $\tilde{Y}^{(k)}$.

Программа 4.58.

```

10 PRINT 'ВЫЧИСЛЕНИЕ МАКСИМАЛЬНОГО СОБСТВЕННОГО ЗНАЧЕНИЯ'
15 PRINT 'МАТРИЦЫ И ПРИНАДЛЕЖАЩЕГО ЕМУ НОРМИРОВАННОГО ВЕКТОРА'
20 INPUT 'ВВЕДИТЕ N, E 'N, E: DIM A(N, N): FOR I=1 TO N: FOR J=1 TO N
30 PRINT 'ВВЕДИТЕ A 'I, 2, 0: I', 'J: INPUT A(I, J): NEXT J: NEXT I
40 FOR I=1 TO N: PRINT I, 2, 0: I 'ВВЕДИТЕ Y0('I) '
50 INPUT A(0, I): NEXT I: LET W=0
60 LET L=0: LET C=0: LET R=0: FOR J=1 TO N: LET S=0
70 FOR I=1 TO N: LET S=S+A(I, J)*A(0, I): NEXT I
80 LET A(J, 0)=S: IF ABS(S)>R THEN LET R=ABS(S)
90 IF S=0 THEN 120
100 IF A(0, J)=0 THEN 120
110 LET C=C+1: LET L=L+A(J, 0)/A(0, J)
120 NEXT J: LET L=L/C
130 FOR J=1 TO N: LET A(0, J)=A(J, 0)/R: NEXT J
140 IF ABS(L-W)>=E THEN LET W=L: GOTO 60
150 PRINT I, 9: 'МАКСИМАЛЬНОЕ СОБСТВЕННОЕ ЗНАЧЕНИЕ L='L
160 PRINT 'ВЕКТОР': FOR I=1 TO N
170 PRINT I, 2, 0: I 'Y 'I ' = ' I, 9: I A(0, I): NEXT I: END
    
```

Пр и м е р. Для контроля этой программы воспользуемся примером из [41]: на куб

твердого материала ρ со всех сторон действующей силы, которые в матричной форме записываются в виде (все элементы матрицы нормированы, реальные усилия получаем их умножением на 10^6 Н/м²)

$$A = \begin{bmatrix} 10 & 5 & 6 \\ 5 & 20 & 4 \\ 6 & 4 & 30 \end{bmatrix}.$$

При $N=3$ и $\varepsilon=E=1 \cdot 10^{-4}$ находим $\mu_{\max} = L_{\max} = 33,709066694$. Кроме того, вычисляется принадлежащий ему вектор: $y_1 = -0,3408143922$, $y_2 = 0,4160822399$ и $y_3 = 1$.

Все действительные собственные значения симметричной матрицы (или заданное их число) могут определяться методом скалярных произведений, реализованным с помощью следующего алгоритма.

1. Вычисляем вектор $Y^{(k)} = A_i Y^{(k-1)}$, где k — номер итерации, A_i — матрица A .

Находим начальные приближения к максимальному собственному значению

$$\mu_k = (y^{(k)}, y^{(k)}) / (y^{(k-1)}, y^{(k-1)}).$$

3. Нормируем вектор $Y^{(k)}$, т. е. заменяем его вектором

$$\tilde{Y}^{(k)} = Y^{(k)} / \sqrt{(y^{(k)}, y^{(k)})}.$$

4. Проверяем выполнение неравенства

$$|\mu_k - \mu_{k-1}| \leq \varepsilon.$$

Если оно выполняется, считаем μ_k наибольшим собственным значением матрицы A_i , а $\tilde{Y}^{(k)}$ — принадлежащим ему собственным вектором. Если данное условие не выполняется, повторяем вычисления с п. 1, приняв за $Y^{(k+1)}$ вектор $\tilde{Y}^{(k)}$.

5. Если задано число искоемых собственных значений $i=1, 2, \dots, k$, то для каждого i повторяем вычисления по п. 1—4, преобразовав матрицу A_i по формуле

$$A_i = A_{i-1} - \lambda_i \{U_i, U_i'\}, \quad i=1, 2, \dots, k-1,$$

где $\{U_i, U_i'\}$ — произведение вектора-столбца на вектор-строку.

Этот метод применим, если все собственные значения матрицы A действительны,

а наибольшие собственные значения не кратно. Перед началом вычислений вводится матрица A , задается погрешность вычислений $\epsilon = E$, число искомых собственных значений и векторов K ($1 \leq K \leq N = n$) и начальный вектор $Y^{(1)} = U$. Для экономии памяти ПЭВМ симметричная матрица задается своим нижним треугольником построчно и занимает одномерный массив с объемом элементов, равным $N(N+1)/2$.

Программа 4.59.

```

10 PRINT 'ВЫЧИСЛЕНИЕ СОБСТВЕННЫХ ЗНАЧЕНИЙ И ПРИНАДЛЕЖАЩИХ ИМ'
15 PRINT 'ВЕКТОРОВ ДЛЯ СИММЕТРИЧНОЙ МАТРИЦЫ МЕТОДОМ СКАЛЯРНЫХ'
20 PRINT 'ПРОИЗВЕДЕНИЙ С ИСЧЕРПЫВАНИЕМ'
25 INPUT 'ВВЕДИТЕ N, E, K: ', N, E, K: LET W=N*(N+1)/2: LET U=W+N: DIM A(U+N)
30 LET H=0: FOR I=1 TO N: FOR J=1 TO I: LET H=H+1
40 PRINT!2.0!'ВВЕДИТЕ A'I', 'J': INPUT A(H),
50 NEXT J:NEXT I:FOR B=1 TO K
60 FOR I=1 TO N:LETA(W+I)=1:NEXT I:LETF=0
70 FOR I=1 TO N:LETA(U+I)=0:FOR J=1 TO I
80 LET A(U+I)=A(U+I)+A(I*(I-1)/2+J)*A(W+I):NEXT J
90 NEXT I:FOR J=1 TO N-1:FOR I=J+1 TO N
100 LET A(U+J)=A(U+J)+A(I*(I-1)/2+J)*A(W+I)
110 NEXT I:NEXT J:LETP=0:LETR=0
120 FOR I=1 TO N:LETP=P+A(U+I)*A(U+I)
130 LETR=R+A(W+I)*A(U+I):NEXT I
140 LETL=P/R:LETC=SQR(P):LETP=0
150 FOR I=1 TO N:LETA(W+I)=A(U+I)/C
160 IF ABS(A(W+I))>P THEN LETP=ABS(A(W+I))
170 NEXT I:IF ABS(P-F)>E THEN LETF=P:GOTO 70
180 PRINT 'СОБСТВЕННОЕ ЗНАЧЕНИЕ L И ЕГО ВЕКТОР V'
190 PRINT!2.0!'L'I'='!F1.9!L
200 FOR I=1 TO N:PRINT!2.0!'V'I'='!F1.9!A(W+I)
210 NEXT I:IF B=K THEN 240
220 FOR I=1 TO N:FOR J=1 TO I:LETD=I*(I-1)/2+J
230 LET A(D)=A(D)-L*A(W+I)*A(W+J):NEXT J:NEXT I
240 NEXT B:END

```

Пример. Для матрицы из примера к программе 4.58 получаем: 1) $L = 33,70917846$.

$y_1 = 0,300158658$, $y_2 = 0,3664623805$,
 $y_3 = 0,8806872904$; 2) $L = 19,14906125$,
 $y_1 = 0,1967307594$, $y_2 = 0,8796183084$,
 $y_3 = -0,4330919532$ и 3) $L = 7,141760291$,
 $y_1 = 0,933376197$, $y_2 = -0,3032741433$,
 $y_3 = -0,19192099959$.

Для отыскания всех собственных значений матрицы применяется также метод Якоби с преградами. Его суть заключается в проведении цепочки преобразований подобия, в ходе которых получается некоторая диагональная матрица $A^{(k)}$: $A^{(k)} = T^k A T^k$, имеющая те же собственные значения, что и матрица A . Матрица T является произведением всех матриц T_{ij} , где T_{ij} — элементарные матрицы вращения, имеющие вид

$$T_{ij} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & C & \dots & S & \\ & & \vdots & \ddots & \vdots & \\ & & & & 1 & \\ & & & & -S & \dots & C \\ & & & & & & 1 \end{bmatrix},$$

$i < j$,
 $C^2 + S^2 = 1$.

Алгоритм реализации метода Якоби с преградами следующий.

1. Задаем $T_{00} = E$, где E — единичная матрица порядка n .

2. Вычисляем первую преграду

$$\alpha_0 = \frac{1}{n} \sqrt{\sum_{i,j=1}^n a_{ij}^2}$$

3. Находим внедиагональный элемент a_{ij} , по модулю превосходящий α_k ($k=0, 1, 2, \dots$). Если его нет, выполняем п. 5 алгоритма, иначе идем к п. 4.

4. Анализирруем элемент a_{ij} . Для этого вычисляем

$$y = (a_{ii} - a_{jj})/2,$$

$$x = \begin{cases} -1, & \text{если } y = 0, \\ -\text{sign}(y) a_{ij} \sqrt{a_{ii}^2 + y^2}, & \text{если } y \neq 0, \end{cases}$$

$$S = x/\sqrt{2(1-\sqrt{1-x^2})}, \quad C = \sqrt{1-S^2}.$$

Затем преобразуем i -й и j -й столбцы матрицы $A^{(k)}$ по формулам

$$B_i = CA_i - SA_j, \quad B_j = SA_i + CA_j$$

и заменяем i -й и j -й столбцы матрицы A столбцами B_i и B_j . В результате на месте матрицы A получаем матрицу B . Далее i -ю и j -ю строки матрицы B преобразуем по формулам

$$C_i = CA_i - SA_j, \quad C_j = SA_i + CA_j$$

и заменяем i -ю и j -ю строки матрицы B строками C_i и C_j . При этом вместо матрицы V получаем матрицу $A^{(k+1)}$. Столбцы i и j матрицы T_{ij} заменяем на столбцы B_i и B_j .

5. Находим новую преграду

$$\alpha_{k+1} = \alpha_k/n^{2(k+1)+1}$$

и повторяем вычисления с п. 1, до тех пор, пока все внедиагональные элементы не станут меньше числа $P = \epsilon \alpha_0$, где ϵ — заданная погрешность вычислений. В результате собствен-

венные значения оказываются диагональными элементами матрицы A .

Ввиду симметрии матрицы она задается построчно нижним треугольником и занимает одномерный массив с числом элементов $N(N+1)/2$.

Программа 4.60.

Пример. Для матрицы A из примера к программе 4.58 вычисления по программе 4.60 дают $L_1=7,14176$, $L_2=19,14906$ и $L_3=33,70917$ ($N=3$, $\epsilon=E=1 \cdot 10^{-5}$).

Программы 4.57—4.60 получены переводом программ для ЭВМ «Мир-2» с языка аналитик [24] на язык бейсик.

```
10 PRINT 'ВЫЧИСЛЕНИЕ ВСЕХ СОБСТВЕННЫХ ЗНАЧЕНИЙ МАТРИЦЫ'
15 PRINT ' МЕТОДОМ ЯКОБИ С ПРЕГРАДАМИ'
20 INPUT 'ВВЕДИТЕ ПОРЯДОК МАТРИЦЫ N=':N:DIM A(N*(N+1)/2)
25 INPUT 'ВВЕДИТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕНИЯ E=':E
30 LETH=0:FOR I=1 TO N:FOR J=1 TO I
40 LETH=H+1:PRINT!3.0!'ВВЕДИТЕ A'I','J':INPUT A(H)
50 NEXT J:NEXT I:LETH=0:FOR P=2 TO N
60 FOR Q=1 TO P-1:LETH=H+2*A*(P*(P-1)/2+Q)^2
70 NEXT Q:NEXT P:LETR=SQR(H):LETA=E*R/N
80 LETR=R/N
90 LETB=0:FOR Q=2 TO N:FOR P=1 TO Q-1
100 IF ABS(A(Q*(Q-1)/2+P))<R THEN 275
110 LETK=P*(P-1)/2+P:LETL=Q*(Q-1)/2+P
120 LETM=Q*(Q-1)/2+Q:LETB=1:LETU=A(K):LETW=A(L)
130 LETF=A(M):LETY=(U-F)/2:LETZ=Y:IF Y=0 THEN LETX=-1:GOTO 150
140 LETX=-SGN(Z)*W/SQR(W*W+Y*Y)
150 LETS=X/SQR(2*(1+SQR(1-X*X))) :LETC=SQR(1-S*S)
160 FOR I=1 TO N:IF I<=P THEN LETY=P*(P-1)/2+I:GOTO 180
170 LETY=I*(I-1)/2+P
180 IF I<=Q THEN LETX=Q*(Q-1)/2+I:GOTO 200
190 LETX=I*(I-1)/2+Q
200 IF I=Q THEN 220
210 LETD=A(Y)*C-A(X)*S
220 LETA(X)=A(Y)*S+A(X)*C:IF I=Q THEN 240
230 LETA(Y)=D
240 NEXT I
250 LETA(K)=U*C+F*S*S-2*W*C*S
260 LETA(M)=V*S*S+F*C*C+2*W*C*S
270 LETA(L)=(U-F)*S*C+W*(C*C-S*S)
275 NEXT P:NEXT Q
280 IF B=1 THEN 90
290 IF R>A THEN LETR=R/N:GOTO 80
300 FOR I=1 TO N:LETH=A(I*(I-1)/2+I)
310 PRINT!2.0!'СОБСТВЕННОЕ ЗНАЧЕНИЕ L'I='!F1.9!H
320 NEXT I:END
```

§ 5.1. Спектральный анализ на основе дискретного преобразования Фурье

Спектром временной зависимости (функции) $y(t)$ называется совокупность ее гармонических составляющих (гармоник), образующих ряд Фурье (см. § 4.11).

Спектральный анализ периодических функций заключается в нахождении коэффициентов a_k, b_k ряда Фурье

$$y(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos 2\pi k f_1 t + b_k \sin 2\pi k f_1 t), \quad (5.1)$$

где f_1 — частота повторения (или частота первой гармоники), k — номер гармоники. Кроме ряда (5.1) широко используется ряд

$$y(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} M_k \cos(2\pi k f_1 t + \varphi_k), \quad (5.2)$$

где

$$M_k = \sqrt{a_k^2 + b_k^2} \quad (5.3)$$

— амплитуда и

$$\varphi_k = -\arctg(b_k/a_k) \quad (5.4)$$

— фаза гармоник (косинусоид). Применяются также ряды с синусоидами под знаком суммы.

Коэффициенты Фурье определяются выражениями

$$a_k = \frac{2}{T} \int_0^T y(t) \cos 2\pi k f_1 t, \quad (5.5)$$

$$b_k = \frac{2}{T} \int_0^T y(t) \sin 2\pi k f_1 t, \quad (5.6)$$

где $T=1/f_1$ — период повторения периодической функции $y(t)$.

Спектральный анализ непериодических (финитных) функций, т. е. функций, полностью определенных на отрезке $[0, t_0]$, заключается в вычислении составляющих их комплексной спектральной плотности

$$S(j\omega) = S_C(\omega) + jS_S(\omega) = S(\omega) e^{j\psi(\omega)},$$

где $\omega = 2\pi f$ — угловая частота,

$$S(\omega) = \sqrt{[S_C(\omega)]^2 + [S_S(\omega)]^2} \quad (5.7)$$

и

$$\varphi(\omega) = -\arctg[S_S(\omega)/S_C(\omega)] \quad (5.8)$$

— модуль спектральной плотности и фаза на частоте ω . При этом

$$S_C = \int_0^{t_0} y(t) \cos \omega t, \quad (5.9)$$

$$S_S = \int_0^{t_0} y(t) \sin \omega t. \quad (5.10)$$

Численный спектральный анализ заключается в нахождении коэффициентов $a_0, a_1, \dots, a_k, b_1, b_2, \dots, b_k$ (или $M_1, M_2, \dots, M_k, \varphi_1, \dots, \varphi_k$) для периодической функции $y(t)$, заданной на отрезке $[0, T]$ дискретными отсчетами. Он сводится к вычислению (5.5) и (5.6) по формулам численного интегрирования для метода прямоугольников (см. § 4.8)

$$a_k = \frac{2}{N} \sum_{i=0}^{N-1} y_i \cos 2\pi k f_1 i \Delta t, \quad (5.11)$$

$$b_k = \frac{2}{N} \sum_{i=0}^{N-1} y_i \sin 2\pi k f_1 i \Delta t; \quad (5.12)$$

где $\Delta t = T/N$ — шаг, с которым расположены абсциссы $y(t)$.

Для финитных функций

$$S_C = \Delta t \sum_{i=0}^{N-1} y_i \cos(2\pi f \Delta t i), \quad (5.13)$$

$$S_S = \Delta t \sum_{i=0}^{N-1} y_i \sin(2\pi f \Delta t i). \quad (5.14)$$

Найденные по (5.11) и (5.12) коэффициенты Фурье для $m=N/2$ гармоник приближают $y(t)$ рядом (5.1) или (5.2) с наименьшей среднеквадратической погрешностью. Следовательно, численный спектральный анализ, совместно со спектральным синтезом (§ 4.11), является разновидностью метода наименьших квадратов, когда $y(t)$ приближается тригонометрическим рядом.

Обобщенный численный спектральный анализ базируется на том, что периодические функции $y(t)$ являются частными случаями случайных финитных функций. Действительно, полагая $t_0 = T$ (этим мы условно приписываем финитным колебаниям период T) и считая $f = k/T$ (при финитных функциях k — любые, а при периодических — целые числа), из (5.11), (5.13) и (5.12), (5.14) получаем

$$A_{k0} = \frac{S_C}{\Delta t} = \frac{a_k N}{2} = \sum_{i=0}^{N-1} y_i \cos 2\pi f_i \Delta t, \quad (5.15)$$

$$B_{k0} = \frac{S_S}{\Delta t} = \frac{b_k N}{2} = \sum_{i=0}^{N-1} y_i \sin 2\pi f_i \Delta t. \quad (5.16)$$

Численный спектральный анализ повышенной точности основан на априорном представлении $y(t)$ в промежутках между узлами. Если такое представление невозможно, используются формулы (5.15) и (5.16), дающие наименьшую среднеквадратичную погрешность (рис. 5.1, а). Если $y(t) = \text{const}$ в промежутках между узлами (рис. 5.1, б),

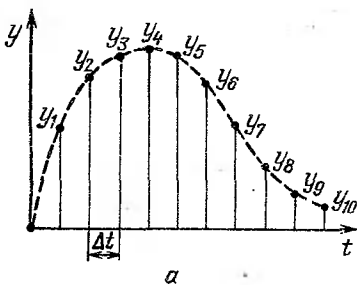


Рис. 5.1. Задание $y(t)$ решетчатой

позволяет уменьшить время анализа, если $y(T) = 0$ в начале или конце отрезка $[0, T]$ или $[0, t_0]$. Однако отсчеты $y_i = 0$ при $N1 \leq i \leq M$ должны вводиться.

3. Задаем шаг Δt .

4. Переменным C и S присваиваем нулевое значение, организуем ввод частоты f и находим $\rho = \pi f \Delta t$.

5. Находим A_0 и B_0 с помощью цикла, в котором вычисляются

$$C = C + y_i \cos(2i\rho), \quad S = S + y_i \sin(2i\rho),$$

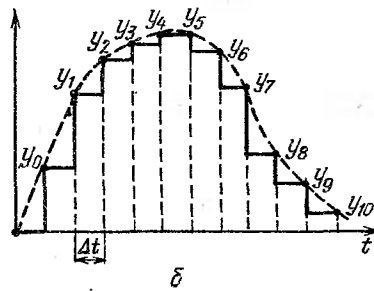
где $i = N1, N1+1, \dots, M$. При выходе из цикла $A_0 = C$ и $B_0 = S$.

6. Для заданного значения f находим нужные параметры a_k, b_k или M_k и φ_k, S_C, S_S или S и φ . Например, для вычисления S и φ применяем формулы

$$S/\Delta t = m = \sqrt{A_0^2 + B_0^2}, \quad (5.20)$$

$$\varphi_{\text{рад}} = Q = -\arccos(A_0/m) \text{ при } B_0 \geq 0, \quad (5.21)$$

$$\varphi_{\text{рад}} = -Q \text{ при } B_0 < 0.$$



(а) и ступенчатой (б) функциям

то можно получить для A_k, B_k точные значения A_{k1}, B_{k1} , вычисляя их по формулам

$$A_{k1} = K_f A_{k0}, \quad B_{k1} = K_f B_{k0}, \quad (5.17)$$

где $K_f = 1$ при $f = 0$,

$$K_f = (\sin \pi f \Delta t) / \pi f \Delta t \text{ при } f \neq 0. \quad (5.18)$$

Формулы (5.17) с учетом K_f (5.18) получаются в результате аналитического интегрирования (5.5) и (5.6). Если в промежутках между узлами функция $y(t)$ аппроксимируется линейной зависимостью, то уточненные значения A и B будут равны

$$A_{k2} = K_f^2 A_{k0}, \quad B_{k2} = K_f^2 B_{k0}. \quad (5.19)$$

Последовательный спектральный анализ на ЭВМ выполняется по следующему алгоритму.

1. Вводим N — число интервалов разбиения $y(t)$, $N1$ — номер первого ненулевого отсчета $y(t)$ и номер последнего ненулевого отсчета M .

2. Организуем цикл ввода ненулевых отсчетов y_i с управляющей переменной I , меняющейся с шагом 1 от значения $N1$ до M .

Формулы (5.21) дают представление $\varphi_{\text{рад}}$ в пределах $\pm \pi$ рад (или $\pm 180^\circ$, если φ рад умножить на множитель $180/\pi$). С помощью (5.17) — (5.19) могут вычисляться уточненные значения $A_1, B_1, S_1 = k_f S$ и $A_2, B_2, S_2 = k_f^2 S$ при ступенчатой или кусочно-линейной аппроксимации $y(t)$.

7. Возвращаемся к п. 4 и повторяем вычисления для нового значения частоты f .

Таким образом, последовательный спектральный анализ обеспечивает возможность вычисления амплитуды и фазы любой гармоники (или спектральной плотности на любой частоте). При этом необходимо запоминание всех отсчетов y_i , кроме нулевых в начале и в конце интервала дискретизации $y(t)$. Число отсчетов ограничено емкостью ОЗУ ПЭВМ и достигает 100—250 у простых ПЭВМ. Из-за резкого увеличения времени вычислений (оно пропорционально N^2) спектральный анализ описанным методом на ПЭВМ при большем числе отсчетов нецелесообразен.


```

05 PRINT 'ПОСЛЕДОВАТЕЛЬНЫЙ СПЕКТРАЛЬНЫЙ АНАЛИЗ'
10 INPUT 'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ РАЗДЕЛЕНИЯ N='N
15 INPUT 'ВВЕДИТЕ НОМЕР ПЕРВОГО НЕНУЛЕВОГО ОТСЧЕТА N1='N1
20 INPUT 'ВВЕДИТЕ НОМЕР ПОСЛЕДНЕГО НЕНУЛЕВОГО ОТСЧЕТА M='M
30 DIM Y(M):FOR I=N1 TO M
35 PRINT I:0:'ВВЕДИТЕ ОТСЧЕТ Y('I')':INPUT Y(I)
40 NEXT I:INPUT 'ЗАДАЙТЕ ШАГ T='T
50 INPUT 'ВВЕДИТЕ ЧАСТОТУ F='F:LETC=0:LETS=0
60 LETP=#PI*F*T:LETM=2*M:FOR I=N1 TO M
70 LETZ=M*I:LETC=C+Y(I)*COS(Z):LETS=S+Y(I)*SIN(Z):NEXT I
80 LETR=SQR(C*C+S*S):LETF=-ACS(C/R)-#PI*T*F
90 IF S<0 THEN LETF=-F
100 LETK=SIN(P)/P:LETR1=K*R:LETR2=K*R*I
110 PRINT!F1.9!'ПРИ РЕШЕТЧАТОЙ Y(T) S(F)='R*T
120 PRINT'ПРИ СТУПЕНЧАТОЙ Y(T) S1(F)='R1*T
130 PRINT'ПРИ КУСОЧНО-ЛИНЕЙНОЙ Y(T) S2(F)='R2*T
140 PRINT'ФАЗОВЫЙ УГОЛ В РАДИАНАХ F='F
150 PRINT'ФАЗОВЫЙ УГОЛ В ГРАДУСАХ F='DEG(F)
160 GOTO 50:END
    
```

Пример. Найти спектральную плотность прямоугольного импульса, заданного отсчетами $y_0=1, y_1=1, \dots, y_7=1$ при $N=32$ и $\Delta t=0,125 \cdot 10^{-6}$ с для частоты $f=250\,000$ Гц. Вводим эти данные ($N1=0, M=7$) и получаем: $S_0=9,01764195 \cdot 10^{-7}$; $S_1=9,003163162 \times 10^{-7}$ (это значение полностью совпадает с теоретическим, поскольку для прямоугольного импульса ступенчатая аппроксимация $y(t)$ является точной); $S_2=8,98870762 \cdot 10^{-7}$ и фазовый сдвиг $\varphi = -39,375^\circ$.

При интерпретации результатов вычисления фазового сдвига следует помнить о конечной области определения углов (в данном случае от -180° до $+180^\circ$). Кроме того, надо учитывать, что значения аргумента у синусов и косинусов не должны выходить за пределы, допустимые для данной ПЭВМ (см. § 2.6). Из-за приближенности аппроксимации $y(t)$ значения φ могут сильно отличаться от точных. Иногда это можно устранить, введя поправку для φ , равную $\Delta\varphi = -\pi f \Delta t$ (в радианах) или $\Delta\varphi = 180 f \Delta t$ (в градусах). Так, в примере к программе 5.1 $\Delta\varphi = 180 \cdot 2,5 \cdot 10^5 \cdot 0,125 \cdot 10^6 = 5,625^\circ$. При этом скорректированное значение угла $\varphi = -(39,375 + 5,625) = -45^\circ$ равно точно ну значению.

Параллельный спектральный анализ на ЭВМ заключается в одновременном (параллельном) вычислении M гармоник. При этом память ЭВМ нужна для запоминания коэффициентов a_k, b_k (или A_k, B_k), однако запоминать все отсчеты y_i не требуется, поскольку каждый отсчет используется для вычисления всех гармоник по мере его ввода. Параллельный спектральный анализ проводится по следующему алгоритму.

1. Вводим нужное число гармоник M , номер начальной гармоники kS , общее число отсчетов $y(t)$ N , номер первого ненулевого отсчета IS и номер последнего ненулевого отсчета IF .

2. Обнуляем переменные массивы $A(k)$ и $B(k)$ и вычисляем $R=2\pi/N$.

3. Организуем цикл ввода $y_i=Y$ с управляющей переменной I , меняющейся с шагом 1 от значения IS до IF . В этом цикле вво-

дим y_i , вычисляем $Z=RI$ и организуем внутренний цикл (п. 4).

4. Организуем цикл вычислений A_k, B_k с управляющей переменной k , меняющейся с шагом 1 от значения 0 до $M-1$. В начале цикла вычисляем $W=Z(k+S)$, где $S=IS$, и затем

$$A(k) = A(k) + Y \cos W,$$

$$B(k) = B(k) + Y \sin W.$$

Значения A_k соответствуют переменным массива $A(k)$, а B_k — переменным массива $B(k)$ после выхода из циклов. Для нулевой гармоники $k+S=0$ вычисляем только A_0 , суммируя все y_i .

5. Организуем цикл вывода A_k, B_k, M_k и $\varphi_k=Q_k$ ($k=0, \dots, M-1$), учитывая сдвиг индекса на величину S , т. е. вывода вместо индекса k значения $k+S$. В ходе вывода вычисляем

$$M_2 = \sum_{i=k+S}^{M+S} M_i^2.$$

6. Если $M1=0$, т. е. вычисления начаты с первой гармоники, находим коэффициент гармоник:

$$k_c = \sqrt{M_2^2 + M_3^2 + \dots + M_M^2} / M_1 = \sqrt{M_2 - M_1^2} / M_1.$$

7. Если $S=1$, можно проводить тригонометрическую интерполяцию — экстраполяцию, т. е. по заданным t вычислять

$$y(t) = \frac{A_0}{2} + \frac{2}{N} \left(\sum_{k=1}^M A_k \cos 2\pi k f \Delta t + \sum_{k=1}^M B_k \sin 2\pi k f \Delta t \right)$$

с помощью усеченного ряда Фурье, ограниченного M гармониками.

Когда по этому алгоритму надо вычислить новый ряд гармоник, приходится повторять

ввод всех отсчетов y_i . Если память ПЭВМ достаточна, можно предусмотреть хранение y_i с помощью массива $Y(I)$, содержащего действующие (т. е. $y_i \neq 0$) отсчеты. Тогда, предусмотрев в п. 3 алгоритма вместо ручного ввода y_i циклический вызов отсчетов из массива $Y(I)$, получим вариант комбинированного или *последовательно-параллельного* спектрального анализа. В приведенной ниже программе 5.2 реализован этот метод. Кроме того, в ней предусмотрен ввод опорной частоты $f_1 = F1$. Это частота первой гармоники при анализе периодических функций или частота, задающая масштаб сетки частот kf_1 при анализе финитных функций (т. е. в последнем случае им придается характер периодических функций с частотой повторения f_1). Коррекция M_k и фазового сдвига в этой программе не предусмотрена (но может быть легко введена).

Программа 5.2.

```

10 PRINT'ПОСЛЕДОВАТЕЛЬНО-ПАРАЛЛЕЛЬНЫЙ СПЕКТРАЛЬНЫЙ АНАЛИЗ'
15 PRINT'И ТРИГОНОМЕТРИЧЕСКАЯ ИНТЕРПОЛЯЦИЯ-ЭКСТРАПОЛЯЦИЯ'
20 INPUT'ЗАДАЙТЕ ОБЩЕЕ ЧИСЛО ОТСЧЕТОВ ФУНКЦИИ Y(T) N='N
30 INPUT'ЗАДАЙТЕ НОМЕРА НАЧАЛЬНОГО И КОНЕЧНОГО ОТСЧЕТОВ IS,IF 'U,B
40 LETG=6-U:DIM Y(G),A(N/2),B(N/2):LETR=2**PI/N
50 FOR I=0 TO G:PRINT!3.0!'ВВЕДИТЕ Y'I+U:INPUT Y(I):NEXT I
60 INPUT'ЗАДАЙТЕ ЧИСЛО СОСТАВЛЯЮЩИХ СПЕКТРА M='M
70 INPUT'ЗАДАЙТЕ НОМЕР ПЕРВОЙ СОСТАВЛЯЮЩЕЙ СПЕКТРА KS='S
75 INPUT'ЗАДАЙТЕ ЧАСТОТУ F1='F
80 LETA=0:LETB=0:LETC=0:FOR K=1 TO M
90 LETA(K)=0:LETB(K)=0:NEXT K
100 FOR I=0 TO G:LETA=A+Y(I):LETB=B*(I+U)
110 FOR K=1 TO M:LETC=C*(K+S-1)
120 LETA(K)=A(K)+Y(I)*COS(W):LETB(K)=B(K)+Y(I)*SIN(W)
130 NEXT K:NEXT I:LETM0=A/N:LETS0=M0/F
140 PRINT!F1.9!'ПОСТОЯННАЯ СОСТАВЛЯЮЩАЯ M0='M0
150 PRINT'СПЕКТРАЛЬНАЯ ПЛОТНОСТЬ НА НУЛЕВОЙ ЧАСТОТЕ S0='S0
160 FOR K=1 TO M:LETU=SQR(A(K)*A(K)+B(K)*B(K))
170 LETQ=-ACS(A(K)/U):IF B(K)<0 THEN LETQ=-Q
180 LETQ=K+S-1:LETB=B+U*U:IF K+S=2 THEN LETC=U
190 PRINT!3.0!'АМПЛИТУДА M'0!F1.9!'='U/N
200 PRINT!3.0!'СПЕКТРАЛЬНАЯ ПЛОТНОСТЬ S'0!F1.9!'='U/N/F
210 PRINT!3.0!'ФАЗОВЫЙ СДВИГ Q'0!F1.9!'='DEG(Q)
220 NEXT K:IFS<>1 THEN 60
230 PRINT'КОЭФФИЦИЕНТ ГАРМОНИК KГ='!F1.9!SQR(B-C**C)/C
240 INPUT'1-БУДЕТ ВЫЧИСЛЯТЬСЯ Y(T),0-НЕ БУДЕТ ?'D
250 IF D<>1 THEN 60
260 INPUT'ЗАДАЙТЕ ВРЕМЯ T='T:LETY=0:LETR=2**PI*T*F
270 FOR K=1 TO M:LETC=C*(K+S-1):LETU=Y+A(K)*COS(W)+B(K)*SIN(W)
280 NEXT K:LETY=M0+Y*2/N:PRINT'ЗНАЧЕНИЕ Y(T)='!F1.9!Y
290 60TO 260:END

```

Пример. Для примера к программе 5.1 ($N=32$, $y_0-y_7=1$, $f_1=250\ 000$), задав $M=5$ и $MS=1$, получим (даны 5 цифр после запятой):

$M_0=0,25$	$S_0=1 \cdot 10^{-6}$
$M_1=0,45088$	$S_1=9,0176 \cdot 10^{-7}$
$M_2=0,32036$	$S_2=6,40729 \cdot 10^{-7}$
$M_3=0,15224$	$S_3=3,0449 \cdot 10^{-7}$
$M_4=2,23345 \cdot 10^{-12}$	$S_4=4,466892 \cdot 10^{-18}$
$M_5=0,09375$	$S_5=1,87503 \cdot 10^{-7}$
$K_r=0,81369$	

$Q_1=-39,375$
 $Q_2=-78,75$
 $Q_3=-118,125$
 $Q_5=-16,875$

Значения M_4 и S_4 недостоверны ввиду их малости (точные значения $M_4=0$ и $S_4=0$). Поэтому случайные значения Q_4 из выдачи исключаются. Для $t=0,5 \cdot 10^{-6}$ по программе 5.2 получим $y(t)=1,07593$, а для $t=3 \cdot 10^{-6}$ получим $y(t)=0,00852$.

Спектральный анализ функций с ограниченным спектром выполняется также непосредственно с помощью дискретного преобразования Фурье над комплексными числами. Так, если задан массив $\dot{u}_i = X_i + jY_i$ ($i=0, 1, 2, \dots, N-1$), то прямое дискретное преобразование Фурье дает массив комплексных чисел

$$\dot{u}_k = \sum_{i=0}^{N-1} \dot{u}_i e^{-j(2\pi ki/N)}$$

Соответственно массив чисел v_k при обратном дискретном преобразовании Фурье обра-

зует массив комплексных чисел

$$\dot{u}_i = \frac{1}{N} \sum_{k=0}^{N-1} \dot{v}_k e^{j(2\pi ki/N)}$$

Если $\dot{u}_i = Y(t_i)$, причем $\text{Re } \dot{u}_i = X_i = Y(t_i)$, а $\text{Im } \dot{u}_i = Y_i = 0$, то прямое преобразование Фурье дает коэффициенты $C_k = A_{k0} - jB_{k0} = v_k/N$ усеченного тригонометрического ряда Фурье в комплексной форме:

$$y(t) = \sum_{k=-\infty}^{\infty} |C_k| e^{j(2\pi k f_1 t + \varphi_k)}$$

где $|C_k| = \sqrt{X_k^2 + Y_k^2}$, $\varphi_k = -\arccos(X_k/|C_k|)$, если $Y_k \geq 0$, и $\varphi_k = -\varphi_k$, если $Y_k < 0$. Пер-

ные $N/2$ значений C_k соответствующих $2M$ раз, поскольку они периодически повторяются. Кроме того, вычисления можно проводить по рекуррентным формулам (см. подробно в [5]). Выделив четную и нечетную части разложения, удается уменьшить время вычислений еще вдвое; в результате БПФ требует проведения $(N/2) \log_2 N$ операций вместо N^2 (при $N=1024$ это уменьшает число операций более чем в 200 раз!). Для ПЭВМ характерны значения $N \leq 64 \div 256$, а для профессиональных ПЭВМ — в несколько раз больше.

Соответственно, имея частотный спектр в виде N значений $C_k = X_k + jY_k$ и подвергнув его обратному преобразованию Фурье, получим N значений $y_i(t_i)/N$. Из них первые $N/2$ значений соответствуют области времен $t \geq 0$, остальные — $t < 0$.

Указанные дискретные преобразования вытекают из прямого преобразования Фурье

$$\hat{S}(\omega) = \int_{-\infty}^{\infty} y(t) e^{-j\omega t} dt$$

и из обратного преобразования Фурье

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{S}(\omega) e^{j\omega t} d\omega,$$

выполняемых методиками приближенного численного интегрирования для функций $y(t)$, определенных на конечном промежутке вре-

мени. Выбор N из ряда 2^M (2, 4, 8, 16 и т. д.) является определенным неудобством БПФ. Однако оно компенсируется заметным уже при $N=8$ или $N=16$ сокращением времени вычислений. Программы БПФ обычно позволяют проводить как прямое, так и обратное дискретное преобразование Фурье над массивами $\dot{u}_i = X_i + jY_i$ и $\dot{v}_k = X_k + jY_k$, обеспечивая преобразование $X_i \rightarrow X_k, Y_i \rightarrow Y_k$, и наоборот.

Программа 5.3.

```

10 PRINT 'ПРЯМОЕ И ОБРАТНОЕ БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ'
20 INPUT 'ДЛЯ N=2^M ВВЕДИТЕ M=M: LET N=INT(2^M+1): DIM X(N-1), Y(N-1)
30 INPUT 'ЗАДАЙТЕ -1 ПРИ ПРЯМОМ БПФ И 1 ПРИ ОБРАТНОМ ' D
35 INPUT 'ВВЕДИТЕ НОМЕРА НАЧАЛЬНОГО И КОНЕЧНОГО ОТСЧЕТОВ IS, IF ' I0, I1
40 FOR I=0 TO I1-I0: PRINT!3.0! 'ДЛЯ I=' I+10
50 INPUT 'ВВЕДИТЕ X I, Y I ' X(I), Y(I): NEXT I
60 FOR L=1 TO M: LET E=INT(2^(M+L)+.1): LET F=E/2: LET U=1: LET V=0
70 LET Z=#PI/F: LET C=COS(Z): LET S=D*SIN(Z): FOR J=1 TO F
80 FOR I=J TO N STEP E: LET O=I-F-1: LET P=X(I-1)+X(O): LET Q=Y(I-1)+Y(O)
90 LET R=X(I-1)-X(O): LET T=Y(I-1)-Y(O): LET X(O)=R*U-T*S
100 LET Y(O)=T*U+R*S: LET X(I-1)=P: LET Y(I-1)=Q: NEXT I
110 LET U=U*C-U*S: LET V=V*C+U*S: LET U=U: NEXT J: NEXT L
120 LET J=1: FOR I=1 TO N-1: IF I>J THEN 150
130 LET J1=J-1: LET I1=I-1: LET P=X(J1): LET Q=Y(J1): LET X(I1)=X(I1)
140 LET Y(I1)=Y(I1): LET X(I1)=P: LET Y(I1)=Q
150 LET K=N/2
160 IF K>J THEN 180
170 LET J=J-K: LET K=K/2: GOTO 160
180 LET J=J+K: NEXT I
190 IF D=-1 THEN PRINT 'РЕЗУЛЬТАТЫ ПРЯМОГО БПФ': GOTO 240
200 PRINT 'РЕЗУЛЬТАТЫ ОБРАТНОГО БПФ': FOR K=0 TO N-1
210 LET X(K)=X(K)/N: LET Y(K)=Y(K)/N
220 PRINT!3.0! 'K=' K: !F1.5! ' X=' X(K): ' Y=' Y(K)
230 NEXT K: STOP
240 FOR K=0 TO N-1: LET A=SQR(X(K)*X(K)+Y(K)*Y(K))
250 LET Q=0: IF A=0 THEN 270
260 LET Q=ACS(X(K)/A): IF Y(K)<0 THEN LET Q=-Q
270 PRINT!3.0! 'K=' K: !F1.5! ' X=' X(K): ' Y=' Y(K):
280 PRINT ' M=' A*M/2: ' Q=' DEG(Q): NEXT K: END

```

мени, и зависимостей $S(\omega)$ с ограниченным спектром.

Дискретное преобразование Фурье требует числа операций порядка N^2 и ведет к большим затратам машинного времени уже при $N > 10 \div 20$. Кроме того (при операциях с комплексными числами), приходится $2N$ раз вычислять тригонометрические функции (\sin и \cos) — операции по разложению их в ряд выполняются на ЭВМ довольно медленно.

Быстрое преобразование Фурье (БПФ) позволяет резко уменьшить время проведения прямого и обратного дискретных преобразований Фурье. При БПФ N выбирается из условия $N=2^M$, где $M=1, 2, 3, \dots$. При этом значения тригонометрических функций доста-

Пример. В программе 5.3 предусмотрен ввод только ненулевых отсчетов. Например, для вычисления спектра прямоугольного импульса, заданного при $N=32$ ($M=5$) восемью единичными отсчетами $Y(t)$ $X_0, X_1, X_2, \dots, X_7$ и остальными нулевыми (все $Y_i=0$), задаем номер начального отсчета $IS=0$ и конечного $IF=7$ и указываем тип преобразования. При прямом дискретном преобразовании Фурье получаем спектральный состав (даны первые 6 значений из таблицы результатов, выводимой на экран дисплея):

K=0	X=8	Y=0
K=1	X=5,57659	Y=-4,57659
K=2	X=1	Y=-5,02734

X=3 X=-1,14828 Y=-2,14828
 K=4 X=0 Y=0
 K=5 X=1,43543 Y=-1,43543

M=0,25 Q=0
 M=0,45088 Q=-39,75
 M=0,32036 Q=-78,75
 M=0,15224 Q=-118,125
 M=0 Q=0
 M=0,09375 Q=-16,875.

Если дать команды D=1 GOTO 60, то после прямого будет проведено обратное дискретное преобразование Фурье. Без учета малых погрешностей округления, получим в итоге значения $X_0 \div X_7 = 1$, $X_8 \div X_{31} = 0$ и $Y_0 \div Y_{31} = 0$ (т. е. исходный массив \dot{u}_i). Время преобразования при M=5 около 2,5 мин.

Повышение точности БПФ достигается специальной обработкой входных или выходных данных. Обработка входных данных заключается в умножении отсчетов на весовые коэффициенты W_i , подобранные так, чтобы обеспечить заданную аппроксимацию $Y(t)$ в ходе интегрирования. Значения $W_i = 1$ соответствуют ступенчатой аппроксимации $Y(t)$ и интегрированию методом прямоугольников. Если $W_i = 1/2, 1, 1, \dots, 1, 1/2$, то $Y(t)$ в ходе БПФ аппроксимируется кусочно-линейной функцией, т. е. интегрирование происходит методом трапеций. При $W_i = 1/3, 4/3, 2/3, 4/3, 2/3, \dots, 4/3, 1/3$ аппроксимация $Y(t)$ будет параболической, т. е. интегрирование происходит по методу Симпсона. При более сложных последовательностях W_i возможна аппроксимация $Y(t)$ полиномами более высоких степеней.

Применим этот метод к примеру для программы 5.3. Зададим 9 отсчетов $X_0 = 1/2$, $X_1 \div X_7 = 1$ и $X_8 = 1/2$ (все $Y_0 \div Y_8 = 0$). С помощью программы 5.3 получим (первые 6 строк таблицы выходных данных):

K=0 X=8 Y=0
 K=1 X=5,07659 Y=-5,07659
 K=2 X=-7.10⁻¹² Y=-5,02734
 K=3 X=1,64828 Y=-1,64828
 K=4 X=0 Y=0
 K=5 X=0,93543 Y=-0,93543

M=0,25 Q=0
 M=0,44871 Q=-45
 M=0,31421 Q=-90
 M=0,14569 Q=-135
 M=0 Q=0
 M=0,09268 Q=-45

При параболической аппроксимации $Y(t)$ задаем ее значениями $X_0 = 1/3$, $X_1 = 4/3$, $X_2 = 2/3$, $X_3 = 4/3$, $X_4 = 2/3$, $X_5 = 4/3$, $X_6 = 2/3$, $X_7 = 4/3$, $X_8 = 1/3$. $X_9 \div X_{31} = 0$ и $Y_0 \div Y_{31} = 0$. Тогда получим:

K=0 X=8 Y=0
 K=1 X=5,0930 Y=-5,0930
 K=2 X=6,6.10⁻¹³ Y=-5,09364
 K=3 X=-1,69884 Y=-1,69884
 K=4 X=0 Y=0
 K=5 X=1,02452 Y=-1,02452

M=0 Q=0
 M=0,45016 Q=-45
 M=0,31835 Q=-90
 M=0,15016 Q=-135
 M=0 Q=0
 M=0,09056 Q=-45

Для данного примера точные результаты вычисляются по формулам

$$X_k = \sin(\pi k/2) N/2\pi k,$$

$$Y_k = -(1 - \cos(\pi k/2)) N/2\pi k,$$

$$M_k = 2 \sqrt{X_k^2 + Y_k^2} / N,$$

$$Q_k = -\arccos(X_k / \sqrt{X_k^2 + Y_k^2})$$

и равны:

K=0 X=8 Y=0
 K=1 X=5,09296 Y=-5,09296
 K=2 X=0 Y=-5,09296
 K=3 X=-1,69765 Y=-1,69765
 K=4 X=0 Y=0
 K=5 X=1,01859 Y=-1,01858

M=0,25 Q=0
 M=0,45016 Q=-45
 M=0,31831 Q=-90
 M=0,15005 Q=-135
 M=0 Q=0
 M=0,09003 Q=-45

Сравнение приведенных данных наглядно иллюстрирует повышение точности БПФ по мере увеличения порядка аппроксимирующего $y(t)$ полинома. Казалось бы, что для прямоугольного импульса ступенчатая аппроксимация будет наилучшей. Однако при БПФ производится приближенное численное интегрирование не самой зависимости $y(t)$, а ее произведения на быстроосциллирующие множители. Поэтому результирующая погрешность интегрирования определяется не только погрешностью аппроксимации $y(t)$, но и всей подынтегральной функции. В следующем разделе описан метод спектрального анализа, в котором по точным формулам интегрируется произведение $y(t)$ на осциллирующие множители (при этом простейшая кусочно-линейная аппроксимация $y(t)$ с разрывами позволяет получить точные результаты спектрального анализа).

Повышение точности БПФ путем обработки выходных данных сводится к умножению X_k и Y_k на множитель $K_k = \sin(\pi k/N) / (\pi k/N)$ и введению поправки на фазовый сдвиг $\Delta\varphi_k = \pi k/N$. В этом случае для рассмотренного примера будем иметь точные значения X_k , Y_k , M_k и φ_k . Разумеется, нельзя использовать одновременно оба способа повышения точности БПФ, поскольку они являются альтернативными вариантами одного и того же подхода — улучшение аппроксимации $Y(t)$.

Эффект Гиббса наблюдается при тригонометрической интерполяции усеченным рядом Фурье функций с разрывами (в частности, прямоугольного импульса). Он заключается в осцилляциях расчетной зависимости ряда Фурье, амплитуда которых может

достигать 18%. Увеличение числа членов ряда ведет лишь к уменьшению длительности осцилляций. Осцилляции значительно снижаются (сглаживаются) при описанной выше обработке данных (как обычного спектрального анализа, так и с помощью БПФ).

§ 5.2. Специальные виды спектрального анализа

Метод пяти ординат обеспечивает вычисление постоянной составляющей Y_0 и первых четырех гармоник $Y_{M1} \div Y_{M4}$ реакции системы с передаточной характеристикой $y(x)$ на гармоническое воздействие $x = x_M \cos \omega_1 t + x_0$ (см. рис. 5.2):

$$Y_0 = [(y_1 + y_5) + 2(y_2 + y_4)] / 6,$$

$$Y_{M1} = (y_1 - y_5 + y_2 - y_4) / 3,$$

$$Y_{M2} = (y_1 + y_5 - 2y_3) / 4,$$

$$Y_{M3} = [(y_1 - y_5) - 2(y_2 - y_4)] / 6,$$

$$Y_{M4} = [(y_1 + y_5) - 4(y_2 + y_4)] / 12,$$

$$K_T = \sqrt{Y_{M2}^2 + Y_{M3}^2 + Y_{M4}^2} / Y_{M1}.$$

Метод широко применяется для расчета радиотехнических систем с малой нелинейностью зависимости $y(x)$.

Программа 5.4.

```

10 PRINT 'СПЕКТРАЛЬНЫЙ АНАЛИЗ МЕТОДОМ 5-ТИ ОРДИНАТ'
20 INPUT 'ВВЕДИТЕ ПЯТЬ ОРДИНАТ ФУНКЦИИ Y1, Y2, Y3, Y4, Y5' Y1, Y2, Y3, Y4, Y5
30 LET Y0=(Y1+Y5+2*(Y2+Y4))/6
40 PRINT 'СРЕДНЕЕ ЗНАЧЕНИЕ Y0=' Y0
50 LET M1=(Y1-Y5+Y2-Y4)/3
60 PRINT 'АМПЛИТУДА ПЕРВОЙ ГАРМОНИКИ M1=' M1
70 LET M2=(Y1+Y5-2*Y3)/4
80 PRINT 'АМПЛИТУДА ВТОРОЙ ГАРМОНИКИ M2=' M2
90 LET M3=(Y1-Y5-2*(Y2-Y4))/6
100 PRINT 'АМПЛИТУДА ТРЕТЬЕЙ ГАРМОНИКИ M3=' M3
110 LET M4=((Y1+Y5)-4*(Y2+Y4)+6*Y3)/12
120 PRINT 'АМПЛИТУДА ЧЕТВЕРТОЙ ГАРМОНИКИ M4=' M4
130 LET K=(SQRT(M2^2+M3^2+M4^2))/M1
140 PRINT 'КОЭФФИЦИЕНТ ГАРМОНИК K=' K
150 GOTO 20 : END

```

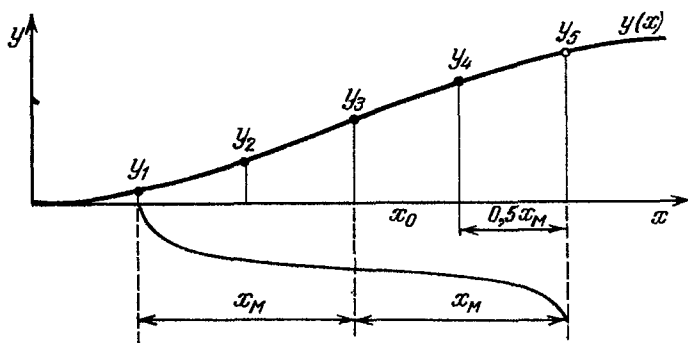


Рис. 5.2. Определение y_i при методе пяти ординат

Пример 6. Для $y_1=0,7$; $y_2=2$; $y_3=3,5$; $y_4=5,2$ и $y_5=6$ получим $Y_0=3,516666667$; $Y_{M1}=-2,833333333$; $Y_{M2}=-0,075$; $Y_{M3}=-0,1833333333$; $Y_{M4}=-0,0916666667$ и $K_T=-0,07703412267$. В программе 5.4 амплитуды гармоник $Y_{M1}-Y_{M4}$ обозначены как $M1 \div M4$.

Метод семи ординат аналогичен, по сути, методу пяти ординат, но отличается большим числом ординат и позволяет найти Y_0 и 6 гармоник [37]:

$$Y_0 = y_4 + (-270\alpha_1 + 756\alpha_2 + 334\alpha_3) / 2560,$$

$$Y_{M1} = (-180\beta_1 + 1008\beta_2 + 668\beta_3) / 2560,$$

$$Y_{M2} = (-1215\alpha_1 + 486\alpha_2 + 559\alpha_3) / 2560,$$

$$Y_{M3} = (-630\beta_1 - 360\beta_2 + 450\beta_3) / 2560,$$

$$Y_{M4} = (270\alpha_1 - 756\alpha_2 + 306\alpha_3) / 2560,$$

$$Y_{M5} = (810\beta_1 - 648\beta_2 + 162\beta_3) / 2560,$$

$$Y_{M6} = (1215\alpha_1 - 486\alpha_2 + 81\alpha_3) / 2560,$$

где

$$\alpha_1 = y_5 + y_3 - 2y_4, \quad \alpha_2 = y_6 + y_2 - 2y_4,$$

$$\alpha_3 = y_7 + y_1 - 2y_4,$$

$$\beta_1 = y_5 - y_3, \quad \beta_2 = y_6 - y_2, \quad \beta_3 = y_7 - y_1,$$

и

$$K_T = \sqrt{Y_{M2}^2 + Y_{M3}^2 + \dots + Y_{M6}^2} / Y_{M1}.$$

Программа 5.5.

```

10 PRINT 'СПЕКТРАЛЬНЫЙ АНАЛИЗ МЕТОДОМ 7-МИ ОРДИНАТ'
20 PRINT 'ВВЕДИТЕ ОРДИНАТЫ Y1, Y2, ..., Y7'
30 INPUT A, B, C, D, E, F, G: LET H=2560
40 LET W=2*M: LET I=E+C-W: LET J=F+B-W
50 LET K=G+A-W: LET L=E-C: LET M=F-B
60 LET N=G-A: LET O=D+(756*J-270*I+334*K)/H
70 LET P=(1008*M-180*L+668*N)/H
80 LET Q=(486*J-1215*I+559*K)/H
90 LET R=(450*N-630*L-360*M)/H
100 LET S=(270*I-756*J+306*K)/H
110 LET T=(810*L-648*M+162*N)/H
120 LET U=(1215*I-486*J+81*K)/H
130 LET V=SQR(Q^2+R^2+S^2+T^2+U^2)/P
140 PRINT 'КОЭФФИЦИЕНТ ГАРМОНИК КГ=' V
150 PRINT 'A0=' O: PRINT 'A1=' P: PRINT 'A2=' Q
160 PRINT 'A3=' R: PRINT 'A4=' S: PRINT 'A5=' T
170 PRINT 'A6=' U: END

```

Пример. Для $y_1=y_2=y_3=0$; $y_4=1$; $y_5=2$; $y_6=3$ и $y_7=4$ получаем $Y_0=1,55625$; $Y_{M1}=2,064375$; $Y_{M2}=0,6265625$; $Y_{M3}=-0,2109375$; $Y_{M4}=-0,05625$; $Y_{M5}=0,1265625$; $Y_{M6}=-0,1265625$ и $K_r=0,3297021676$. В программе 5.5 $Y_0 \div Y_{M6}$ обозначены $A_0 \div A_6$.

Метод степенного полинома основан на аппроксимации передаточной характеристики полиномом

$$y(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

и вычисления спектра $y(t)$ при воздействии в виде $x(t) = x_0 + x_M \cos \omega t$. При этом [37]

$$y_0(t) = a_n; \quad y_1(t) = a_{n-1} + x(t) a_n; \dots;$$

$$y_k(t) = a_{n-k} + x(t) \sum_{i=0}^{k-1} A_i^{(k-1)} \cos i\omega t =$$

$$= \sum_{i=0}^k A_i^{(k)} \cos i\omega t.$$

Коэффициенты $A_k^{(k)}$ последней формулы вычисляются по рекуррентным соотношениям в следующей последовательности:

$$A_0^{(k)} = a_{n-k} + x_M A_1^{(k-1)} / 2;$$

$$A_1^{(k)} = x_M (A_1^{(k-1)} + A_2^{(k-1)}) / 2;$$

$$A_2^{(k)} = x_M (A_1^{(k-1)^2} + A_3^{(k-1)}) / 2; \dots;$$

$$A_i^{(k)} = x_M (A_i^{(k-1)} + A_{i+1}^{(k-1)}) / 2; \dots;$$

$$A_{k-1}^{(k)} = x_M A_{k-2}^{(k-1)} / 2;$$

$$A_k^{(k)} = x_M A_{k-1}^{(k-1)} / 2;$$

Ввиду четности функции $x(t)$ независимо от вида нелинейности $y(x)$ функция $y(t)$ также является четной и ее спектр содержит только косинусоидальные составляющие с фазой $\varphi_i=0$ (и постоянную составляющую $A_0/2$). Число гармонических составляющих равно n , а коэффициент гармоник

$$K_r = \sqrt{A_2^2 + A_3^2 + \dots + A_n^2} / A_1.$$

Программа 5.6.

```

10 PRINT 'АНАЛИЗ РЕАКЦИИ НА ГАРМОНИЧЕСКОЕ ВОЗДЕЙСТВИЕ'
20 PRINT ' НЕЛИНЕЙНОЙ СИСТЕМЫ С ПОЛИНОМИАЛЬНОЙ'
30 PRINT ' ПЕРЕДАТОЧНОЙ ХАРАКТЕРИСТИКОЙ'
40 INPUT 'ВВЕДИТЕ СТЕПЕНЬ ПОЛИНОМА N=' N
50 DIM D(N+1, N+1), A(N)
60 PRINT 'ВВЕДИТЕ КОЭФФИЦИЕНТЫ ПОЛИНОМА'
70 FOR I=0 TO N: PRINT I, 0: A(I): INPUT A(I): NEXT I
80 PRINT 'ВВЕДИТЕ ПОСТОЯННУЮ СОСТАВЛЯЮЩУЮ ВХОДНОГО'
90 INPUT 'ВОЗДЕЙСТВИЯ X0=' X0
100 INPUT 'ВВЕДИТЕ АМПЛИТУДУ ВХОДНОГО ВОЗДЕЙСТВИЯ XM=' XM
110 FOR I=0 TO N+1: FOR J=0 TO N+1
120 LET D(I, J)=0: NEXT J: NEXT I
130 LET D(0, 0)=A(N): FOR I=1 TO N
140 LET D(I, 0)=A(N-I)+X0*D(I-1, 0)+XM*D(I-1, 1)/2
150 LET D(I, 1)=X0*D(I-1, 1)+XM*(D(I-1, 0)+D(I-1, 2))/2
160 FOR J=2 TO N
170 LET D(I, J)=X0*D(I-1, J)+XM*(D(I-1, J-1)+D(I-1, J+1))/2
180 NEXT J: NEXT I
190 LET K=0: FOR J=2 TO N
200 LET K=K+D(N, J)*D(N, J): NEXT J
210 LET K=SQR(K/(D(N, 1)*D(N, 1)))
220 PRINT 'СПЕКТР ВХОДНОГО СИГНАЛА'
230 PRINT 'ПОСТОЯННАЯ СОСТАВЛЯЮЩАЯ Y0=' IF1.9 I D(N, 0)
240 PRINT 'АМПЛИТУДЫ ГАРМОНИК'
250 FOR J=1 TO N: PRINT I, 2, 0: 'YM' J: ' IF1.9 I D(N, J): NEXT J
260 PRINT IF1.9 I: 'КОЭФФИЦИЕНТ ГАРМОНИК КГ=' K
270 GOTO 80: END

```

Примеч. Для $y(x) = 1 + 1000x + 200x^2 + 100x^3 + 20x^4 + 10x^5$; $x_0 = 0$ и $x_M = 4$ получим $A_0 = 3521$; $A_1 = 15200$; $A_2 = 4160$; $A_3 = -4800$; $A_4 = 640$; $A_5 = 640$ и $K_r = 0,42210395$. Амплитуды $A_0 \div A_N$ в программе обозначены $Y_0, Y_{M1} \div Y_{MN}$.

Метод двенадцати ординат является простейшей разновидностью БПФ и служит для вычисления коэффициентов Фурье $b_0 \div b_6$ и $a_1 \div a_5$ функции $y(x)$, заданной 12 равноотстоящими ординатами $y_1 \div y_{12}$. Алгоритм расчета по этому методу следующий [1].

1. Вычисляем суммы и разности ординат по схеме (s_i — суммы, d_i — разности):

	y_1	y_2	y_3	y_4	y_5	y_6
y_{12}	y_{11}	y_{10}	y_9	y_8	y_7	y_6
s_0	s_1	s_2	s_3	s_4	s_5	s_6
	d_1	d_2	d_3	d_4	d_5	

2. Вычисляем суммы σ_i и разности δ_i :

s_0	s_1	s_2	s_3
s_6	s_5	s_4	s_3
σ_0	σ_1	σ_2	σ_3
δ_0	δ_1	δ_2	

3. Вычисляем суммы θ_i и разности η_i :

d_1	d_2	d_3
d_6	d_4	
θ_1	θ_2	θ_3
η_1	η_2	

Находим коэффициенты $b_0 \div b_6$ и $a_1 \div a_5$ ($a_6 = 0$):

$$a_1 = \frac{\theta_1/2 + \theta_3 + \theta_2 \sqrt{3}/2}{6};$$

$$a_2 = \frac{(\eta_1 + \eta_2) \sqrt{3}/2}{6}; \quad a_3 = \frac{\theta_1 - \theta_3}{6};$$

$$a_4 = \frac{\sqrt{3}}{2} (\eta_1 - \eta_2); \quad a_5 = \frac{\theta_1/2 + \theta_3 - \theta_2 \sqrt{3}/2}{6};$$

$$b_0 = \frac{\sigma_0 + \sigma_1 + \sigma_2 + \sigma_3}{12};$$

$$b_1 = \frac{\delta_2/2 + \delta_0 + \delta_1 \sqrt{3}/2}{6};$$

$$b_2 = \frac{\sigma_0 - \sigma_2/2 + \sigma_1/2 - \sigma_3}{6};$$

$$b_3 = \frac{\delta_0 - \delta_2}{6}; \quad b_4 = \frac{\sigma_0 - \sigma_2/2 - \sigma_1/2 + \sigma_3}{6};$$

$$b_5 = \frac{\delta_2/2 + \delta_0 - \delta_1 \sqrt{3}/2}{6};$$

$$b_6 = \frac{\sigma_0 + \sigma_2 - (\sigma_1 + \sigma_3)}{12}.$$

Программа 5.7.

```

10 PRINT 'ГАРМОНИЧЕСКИЙ АНАЛИЗ МЕТОДОМ 12-ТИ ОРДИНАТ'
20 DIM Y(12):FOR I=1 TO 12
40 PRINT!2.0!'ВВЕДИТЕ Y('I)':INPUT Y(I):NEXT I
50 LET S0=Y(12):LET S1=Y(1)+Y(11):LET S2=Y(2)+Y(10)
60 LET S3=Y(3)+Y(9):LET S4=Y(4)+Y(8):LET S5=Y(5)+Y(7):LET S6=Y(6)
90 LET D1=Y(1)-Y(11):LET D2=Y(2)-Y(10):LET D3=Y(3)-Y(9)
100 LET D4=Y(4)-Y(8):LET D5=Y(5)-Y(7)
130 LET G0=S0+S6:LET G1=S1+S5:LET G2=S2+S4:LET G3=S3
140 LET T0=S0-S6:LET T1=S1-S5:LET T2=S2-S4
170 LET O1=D1+D5:LET O2=D2+D4:LET O3=D3
180 LET H1=D1-D5:LET H2=D2-D4
210 LET L=SQR(3)/2:LET A1=((O1/2)+O3+O2*L)/6
220 LET A2=(H1+H2)*L/6:LET A3=(O1-O3)/6
230 LET A4=(H1-H2)*L/6:LET A5=((O1/2)+O3-O2*L)/6
240 LET B0=(G0+G2+G1+G3)/12:LET B1=((T2/2)+T0+T1*L)/6
250 LET B2=(G0-(G2/2)+(G1/2)-G3)/6:LET B3=(T0-T2)/6
260 LET B4=(G0-(G2/2)-(G1/2)+G3)/6
270 LET B5=((T2/2)+T0-T1*L)/6
280 LET B6=(G0+G2-G1-G3)/12
290 PRINT!F1.9!'ЗНАЧЕНИЯ КОЭФФИЦИЕНТОВ'
300 PRINT 'A1='A1:PRINT 'A2='A2:PRINT 'A3='A3
310 PRINT 'A4='A4:PRINT 'A5='A5:PRINT 'B0='B0
320 PRINT 'B1='B1:PRINT 'B2='B2:PRINT 'B3='B3
330 PRINT 'B4='B4:PRINT 'B5='B5:PRINT 'B6='B6
340 GOTO 20:END

```

Примере. Для $y_1=1$; $y_2=4$; $y_3=5$; $y_4=4$; $y_5=3$; $y_6=3,5$; $y_7=3,5$; $y_8=3$; $y_9=1$; $y_{10}=0,5$; $y_{11}=0,2$ и $y_{12}=0$ получим $a_1=1,34118572$; $a_2=0,5484827557$; $a_3=-0,6166666667$; $a_4=-0,1732050808$; $a_5=0,04214761383$; $b_0=2,3916666667$; $b_1=-1,556655773$; $b_2=-0,7333333333$; $b_3=-0,1666666667$; $b_4=-0,0166666667$; $b_5=-0,02667755999$; $b_6=0,1083333333$.

Отметим, что в отличие от введенных в § 5.1 обозначений здесь использованы обратные (a_i обозначено как b_i , а b_i как a_i), что связано с применением ряда Фурье, у которого члены ряда — синусы, а не косинусы.

Последовательный спектральный анализ кусочно-линейных функций с разрывами обеспечивает точное вычисление спектров функций, которые могут быть синтезированы из треугольников, прямоугольников и трапеций. При этом каждый отсчет y_i задается моментом времени t_i и двумя значениями: $y_i(t_i-0)$ слева и $y_i(t_i+0)$ справа, где 0 понимается как бесконечно малое приращение t_i . Если $y_i(t_i-0)=y_i(t_i+0)$, то в точке t_i зависимость $y(t_i)$ не имеет разрыва. Если имеется разрыв, он задается соответствующими значениями функции до и после разрыва. Например, прямоугольный импульс единичной амплитуды задается значениями $t_1, y_1(t_1-0)=0, y_1(t_1+0)=1, t_2, y_2(t_2-0)=1$ и $y_2(t_2+0)=0$. Нумерация отсчетов i далее идет с 1, число отсчетов $y(t)$ равно N . Отсчеты могут идти с произвольным расположением точек t_i на оси времени.

На произвольном участке $[t_{i-1}, t_i]$ функция $y(t)$ аппроксимируется прямой, уравнение которой

$$\frac{y(t) - y_{i-1}}{y_i - y_{i-1}} = \frac{t - t_{i-1}}{t_i - t_{i-1}}$$

Следовательно, взяв отсчеты с соответствующей стороны, получим для $y(t)$ в ин-

тервале $(t_{i-1}+0, t_i-0)$ следующее выражение:

$$y(t) = \left[y(t_{i-1}+0) - \frac{y(t_i-0) - y(t_{i-1}+0)}{t_i - t_{i-1}} \right] + \frac{y(t_i-0) - y(t_{i-1}+0)}{t_i - t_{i-1}} t$$

Подставляя это выражение в (5.11) и (5.12), получим ($x = \pi/k \Delta t$):

$$G_0 = \frac{a_0}{2} = \frac{1}{2N} \sum_{i=1}^N \{y(t_{i-1}+0) + y(t_i-0)\},$$

$$A_k = \frac{1}{xN} \sum_{i=1}^N \{y(t_i-0) \sin(2xi) - y(t_{i-1}+0) \sin(2x(i-1))\} - \frac{\sin x}{Nx^2} \sum_{i=1}^N \{(y(t_i-0) - y(t_{i-1}+0)) \sin(x(2i-1))\},$$

$$B_k = \frac{1}{xN} \sum_{i=1}^N \{y(t_{i-1}+0) \cos(2x(i-1)) - y(t_i-0) \cos(2xi)\} + \frac{\sin x}{x^2 N} \sum_{i=1}^N \{(y(t_i-0) - y(t_{i-1}+0)) \cos(x(2i-1))\}.$$

Алгоритм последовательного спектрального анализа для данного случая подобен описанному в § 5.1.

Программа 5.8.

```

10 PRINT 'ПОСЛЕДОВАТЕЛЬНЫЙ СПЕКТРАЛЬНЫЙ АНАЛИЗ КУСОЧНО-ЛИНЕЙНОЙ'
20 PRINT '    ФУНКЦИИ ЗАДАННОЙ N ОТСЧЕТАМИ, НАЧИНАЯ С I=0'
30 PRINT '    С ПРОИЗВОЛЬНЫМ РАСПОЛОЖЕНИЕМ ОТСЧЕТОВ'
40 INPUT 'ВВЕДИТЕ ОБЩУЮ ДЛИТЕЛЬНОСТЬ ВРЕМЕННОГО ИНТЕРВАЛА T=' T
50 INPUT 'ЗАДАЙТЕ ЧИСЛО ОТСЧЕТОВ N=' N: DIM U(N), Y(1,N)
60 FOR I=1 TO N: PRINT I, 3.0: 'ВВЕДИТЕ T I-1: INPUT U(I)
70 INPUT 'ВВЕДИТЕ ЗНАЧЕНИЯ Y(T-0), Y(T+0) ' Y(0, I), Y(1, I): NEXT I
90 INPUT 'ВВЕДИТЕ ЧАСТОТУ F=' F: IF F > 0 GOTO 150
100 LET A=0: FOR I=2 TO N
110 LET A=A+(U(I)-U(I-1))*Y(1, I-1)+Y(0, I))/(2*T): NEXT I
120 PRINT I, 9.1: 'ПОСТОЯННАЯ СОСТАВЛЯЮЩАЯ M(0)=' A: LET S=T*ABS(A)
130 PRINT 'СПЕКТРАЛЬНАЯ ПЛОТНОСТЬ S(0)=' S: GOTO 90
150 LET A=0: LET B=0: LET R=0: LET X0=#PI*F*T: FOR I=2 TO N
160 LET X1=2*#PI*F*U(I-1): LET X2=2*#PI*F*U(I)
170 LET X3=(Y(0, I)-Y(1, I-1))/(X2-X1)
175 LET D=F*U(I-1): IF D-INT(D)<1.E-5 THEN LET X1=0
180 LET D=F*U(I): IF D-INT(D)<1.E-5 THEN LET X2=0
185 LET S1=SIN(X1): LET S2=SIN(X2): LET C1=COS(X1): LET C2=COS(X2)
190 LET A=A+(X3*(C2-C1)+Y(0, I)*S2-Y(1, I-1)*S1)/X0
200 LET B=B+(X3*(S2-S1)-Y(0, I)*C2+Y(1, I-1)*C1)/X0
210 NEXT I: LET G=SQR(A*A+B*B): IF G=0 GOTO 230
215 LET Z=A/G: IF ABS(Z)<1.E-11 THEN LET Z=0
220 LET R=-ACS(Z): IF B<0 THEN LET R=-R
230 PRINT I, 9.1: '    АМПЛИТУДА            M(F)=' G
240 PRINT '    СПЕКТРАЛЬНАЯ ПЛОТНОСТЬ S(F)=' G*T/2
250 PRINT '    ФАЗА В ГРАДУСAX            Q(F)=' DEG(R): GOTO 90: END

```


Пример 5.9. Для программы с программой 5.1 (вычисление спектра прямоугольного импульса) вводим $T=4 \cdot 10^{-6}$ с, $N=32$, $t_1=0$, $F(t=0)=0$, $F(t+0)=1$, $t_2=1 \cdot 10^{-6}$ с, $F(t=0)=1$, $F(t+0)=0$, $f=250000$. Получим $G(0)=0,25$, $S(0)=1 \cdot 10^{-6}$, $G(f)=0,4501581581$, $S(f)=9,003163162 \cdot 10^{-1}$ и $\varphi(f)=-45^\circ$ (эти значения совпадают с точными). В этой программе нумерация отсчетов i идет с 1, $y(t) \equiv F(T)$, $M(f) \equiv G(F)$, $S(f) \equiv S(F)$ и $\varphi(f) \equiv \Phi(F)$.

Параллельный спектральный анализ кусочно-линейной функции с разрывами основан на применении описанной выше методики при одновременном вычислении рядов a_n , b_n .

Программа 5.9.

```

10 PRINT 'ПАРАЛЛЕЛЬНЫЙ СПЕКТРАЛЬНЫЙ АНАЛИЗ И ТРИГОНОМЕТРИЧЕСКАЯ'
20 PRINT 'ИНТЕРПОЛЯЦИЯ КУСОЧНО-ЛИНЕЙНОЙ ФУНКЦИИ С РАЗРЫВАМИ ПРИ'
30 PRINT '    ПРОИЗВОЛЬНОМ РАСПОЛОЖЕНИИ ОТСЧЕТОВ'
40 PRINT '
50 INPUT 'ЗАДАЙТЕ КРАТНОСТЬ ЧАСТОТ M='M: DIM A(M), B(M)
60 INPUT 'ЗАДАЙТЕ ЧАСТОТУ F='F
70 FOR K=0 TO M: LET A(K)=0: LET B(K)=0: NEXT K
80 LET X=2*PI*F: INPUT 'ЗАДАЙТЕ ДЛИТЕЛЬНОСТЬ ВСЕГО ИНТЕРВАЛА T='T
90 LET Z=X*T/2: INPUT 'ЗАДАЙТЕ ЧИСЛО ОТСЧЕТОВ N='N
100 INPUT 'ВВЕДИТЕ МОМЕНТ ВРЕМЕНИ T O='T0
110 INPUT 'ВВЕДИТЕ ЗНАЧЕНИЕ ФУНКЦИИ F(T O+0)='Y0
120 FOR I=1 TO N-1: PRINT 'ВВЕДИТЕ МОМЕНТ ВРЕМЕНИ T 'I:3.0:I
130 INPUT I: INPUT 'ВВЕДИТЕ ЗНАЧЕНИЯ F(T-O), F(T+O) 'Y1,Y2
140 LET Y=Y1-Y0: FOR K=1 TO M: LET X0=X*K*T0: LET X1=X*K*T1
150 LET X2=Y/(X1-X0): LET D=Z*K: LET S0=SIN(X0)
160 LET S1=SIN(X1): LET C0=COS(X0): LET C1=COS(X1)
170 LET A(K)=A(K)+(Y1*S1-Y0*S0+X2*(C1-C0))/D
180 LET B(K)=B(K)+(Y0*C0-Y1*(C1+X2*(S1-S0))/D: NEXT K
190 LET A(0)=A(0)+(T1-T0)*(Y0+Y1)/2: LET Y0=Y2
200 LET T0=T1: NEXT I: LET E=0
210 PRINT '    K      МОДУЛЬ M(K)      СПЕКТР. ПЛОТН. S(K)      ФАЗА Q(ГРАД.)'
220 PRINT '3.0:0;' 'IF1.5:1:A(0);' 'A(0)*T
230 LET E=0: FOR K=1 TO M: LET G=SGR(A(K)*A(K)+B(K)*B(K))
235 IF K=1 THEN LET G1=G
240 LET E=E+G: LET Q=0: IF G=0 THEN 260
250 LET Q=-ACS(A(K)/G): IF B(K)<0 THEN LET Q=-Q
260 PRINT '3.0:K;' 'IF1.5:1:G;' 'G*T/2;' 'DEG(Q): NEXT K
265 PRINT 'КОЭФФИЦИЕНТ ГАРМОНИК KG='SGR(E-G1*G1)/G1
270 PRINT: INPUT 'ВВЕДИТЕ ВРЕМЯ TX='U: LET Y=A(0): LET P=X*U
280 FOR K=1 TO M: LET F=X*K: LET Y=Y+A(K)*COS(F)+B(K)*SIN(F): NEXT K
290 PRINT 'ЗНАЧЕНИЕ Y(TX)='Y: GOTO 270: END

```

Эту программу можно проверить по примеру, приведенному к программам 5.1 и 5.2. Программа 5.9 обеспечивает одновременно и тригонометрическую интерполяцию, т. е. вычисление $y(t)$ для любого заданного t с помощью гармонического синтеза.

Расчет амплитудно-частотной (АЧХ) и фазо-частотной (ФЧХ) характеристик линейной системы по ее переходной характеристике (ПХ) $a(t)$ является одним из важных практических приложений спектрального анализа. Он основан на следующей связи этих характеристик:

$$A(j\omega) = a(0) + A_C(\omega) + jA_S(\omega),$$

где $A_C(\omega) = \int_0^{\infty} a'(t) \cos \omega t dt$, $A_S(\omega) = \int_0^{\infty} a'(t) \sin \omega t dt$, причем АЧХ ($f = \omega/2\pi$)

$A(f) = \sqrt{A_C^2(f) + A_S^2(f)}$, а ФЧХ $\varphi(f) = -\arctg [A_S(f)/A_C(f)]$. Здесь $a'(t)$ — производная ПХ $a(t)$, $a(0)$ значение $a(t)$ при $t=0$.

Численный метод расчета АЧХ и ФЧХ по ПХ базируется на кусочно-линейной аппроксимации $a(t)$ в интервале между ее отсчетами y_i . Тогда

$$a'(t) \approx \frac{\Delta a(t)}{\Delta t} \approx \frac{a_i - a_{i-1}}{\Delta t}.$$

При этом $a'(t)$ представляется ступенчатой линией, смещенной на $+\Delta t/2$, что создает значительные погрешности в вычислении ФЧХ. Сместив эту линию на $-\Delta t/2$, можно значительно уменьшить эти погрешности. Далее,

применяя для вычисления $A_C(\omega)$ и $A_S(\omega)$ метод численного интегрирования с поправочными коэффициентами (см. § 5.1), получим [9, 10]:

$$A_C(f) = \frac{\sin(\pi f \Delta t)}{\pi f \Delta t} \sum_{i=1}^N (a_i - a_{i-1}) \cos(\pi f \Delta t (2i-1)),$$

$$A_S(f) = \frac{\sin(\pi f \Delta t)}{\pi f \Delta t} \sum_{i=1}^N (a_i - a_{i-1}) \sin(\pi f \Delta t (2i-1)).$$

По вводимым отсчетам ПХ $a_i = a(t_i)$ находим $A_C(\omega)$, $A_S(\omega)$ для заданных f и Δt , после чего находим точки АЧХ $A(f)$ и ФЧХ $\varphi(f)$.

```

05 PRINT 'ВЫЧИСЛЕНИЕ АЧХ И ФЧХ ПО ПК'
10 INPUT 'ЧИСЛО ОТСЧЕТОВ ПК M=' M
20 INPUT 'A=' X: LETM=M-1: DIM A(M)
30 FOR I=1 TO M: INPUT 'A=' Y
40 LET A(I)=Y-X: LET X=Y
50 NEXT I: INPUT 'ШАГ T=' T
60 LET C=0: LET S=0
70 INPUT 'ЧАСТОТА F=' F
75 LET P=(#PI)*F*T
80 FOR I=1 TO M
90 LET C=A(I)*COS(P*(2*I-1))+C
100 LET S=A(I)*SIN(P*(2*I-1))+S: NEXT I
110 LET C1=C*SIN(P)/P: LET S1=S*SIN(P)/P
120 PRINT 'КОЭФ. ПЕРЕДАЧИ ФАЗОВЫЙ УГОЛ'
130 PRINT '!2.3! SQR(C1^2+S1^2), DEG(-ATN(S/C))
140 GOTO 60: END

```

Пример. Для $a_1=0,05$; $a_2=0,4$; $a_3=0,75$; $a_4=1$; $a_5=1,16$; $a_6=1,2$; $a_7=1,16$; $a_8=1,08$; $a_9=1,02$; $a_{10}=1,0$ и $\Delta t=0,1$ получим $A(2)=0,2367203833$, $\varphi(2)=-160,4352421^\circ$.

Метод Берга служит для расчета спектра усеченных косинусов, получаемых при воздействии $x(t)=x_M \cos(\omega t) + X_0$, действующем на входе системы с передаточной характеристикой $y=f(x)$, аппроксимированной двумя отрезками прямых:

$$y(x) = 0 \quad \text{при } x \leq x_0,$$

$$y(x) = Sx \quad \text{при } x > x_0,$$

где S — крутизна передаточной характеристики. При этом косинусоидальный импульс характеризуется амплитудой y_M и углом отсечки

$$\theta = \arccos [(x_0 - X_0)/x_M].$$

Метод Берга сводится к расчетам по формулам

$$a_0 = y_0/y_M = (\sin \theta - \theta \cos \theta)/\pi (1 - \cos \theta),$$

$$a_1 = y_{M1}/y_M = (\theta - \sin \theta \cos \theta)/\pi (1 - \cos \theta),$$

$$a_k = \frac{y_{Mk}}{y_M} = \frac{2 \sin k\theta \cos \theta - k \cos k\theta \sin \theta}{\pi (k^2 - 1) (1 - \cos \theta)}.$$

Программа 5.11.

```

10 PRINT 'РАСЧЕТ КОЭФФИЦИЕНТОВ БЕРГА'
20 PRINT 'ВВЕДИТЕ КОД N'
30 PRINT '0, ЕСЛИ УГОЛ ОТСЕЧКИ НЕ ЗАДАЕТСЯ'
40 INPUT '1, ЕСЛИ УГОЛ ОТСЕЧКИ ЗАДАЕТСЯ N=' N
50 IF N=0 GOTO 80
60 IF N=1 GOTO 125
70 PRINT 'КОД НАБРАН НЕВЕРНО': GOTO 15
80 INPUT 'ЗАДАЙТЕ АМПЛИТУДУ A=' A
90 INPUT 'ЗАДАЙТЕ ПОСТОЯННЫЙ СДВИГ F=' F
100 INPUT 'ЗАДАЙТЕ ПОРОГ X0=' X
110 LET Q=ACS((X-F)/A)
120 PRINT '!1.9! УГОЛ ОТСЕЧКИ В ГРАДУСАХ Q=' DEG(Q): GOTO 130
125 INPUT 'ВВЕДИТЕ УГОЛ ОТСЕЧКИ В ГРАДУСАХ Q=' Q: LET Q=RAD(Q)
130 LET Y0=(SIN(Q)-Q*COS(Q))/(#PI*(1-COS(Q)))
140 PRINT 'КОЭФФИЦИЕНТ A0=' Y0
150 LET Y1=(Q-SIN(Q)*COS(Q))/(#PI*(1-COS(Q)))
160 PRINT 'КОЭФФИЦИЕНТ A1=' Y1
170 INPUT 'ЗАДАЙТЕ НОМЕР ВЫСШЕЙ ГАРМОНИКИ K>2 K=' K
180 LET Y=2*(SIN(K*Q)*COS(Q)-K*COS(K*Q)*SIN(Q))
190 LET Y=Y/(#PI*K*(K^2-1)*(1-COS(Q)))
200 PRINT '!2.0! КОЭФФИЦИЕНТ A' K '=' !1.9! Y: GOTO 170: END

```

Пример. Для заданной амплитуды $x_M=A=10$, сдвига $X_0=F=0$ и порога $x_0=-X_0=0$ получим угол $\theta=Q=90^\circ$, $a_0=-A_0=0,3183098862$, $a_1=A_1=0,5$ и $a_2=-A_2=0,2122065908$:

§ 5.3. Статистический анализ и подготовка гистрограмм

Одномерный массив из N некоторых цифровых данных x_i характеризуется совокупностью статистических характеристик (одномерная статистика), перечисленных ниже.

Начальные моменты k -го порядка

$$m_k(x) = \frac{1}{N} \sum_{i=1}^N x_i^k.$$

Обычно $k=1, 2, 3$ и 4 (точность вычисления m_k при $k > 4$ низкая).

Центральные моменты k -го порядка

$$M_k(x) = \frac{1}{N} \sum_{i=1}^N [x_i - m_1(x)]^k.$$

Момент $M_1(x) = 0$.

Связь центральных моментов с начальными устанавливается соотношениями (аргумент x в скобках опускаем)

$$M_2 = m_2 - m_1^2, \quad M_3 = m_3 - 3m_1m_2 + 2m_1^3,$$

$$M_4 = m_4 - 4m_1m_3 + 6m_1^2m_2 - 3m_1^4,$$

которые позволяют вычислять M_k по мере ввода x_i (без запоминания массива x_i).

Среднее значение

$$\bar{x} = m_1 = \frac{1}{N} \sum_{i=1}^N x_i,$$

есть наиболее вероятное значение числа в массиве.

Дисперсия смещенная

$$D = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

является наиболее вероятной степенью отклонения x_i от среднего значения \bar{x} .

Стандартное отклонение смещенное $\sigma = \sqrt{D}$ определяет среднеквадратичную погрешность x_i , если за точное значение принять \bar{x} .

Дисперсия несмещенная

$$D_0 = M_2 N / (N - 1) = \frac{1}{N - 1} \sum_{i=1}^N (x_i - \bar{x})^2$$

применяется при статистической обработке чисел x_i с нормальным распределением.

Стандартное отклонение несмещенное $\sigma_0 = \sqrt{D_0}$.

вытянут правый, а при $A < 0$ — левый участок спада кривой $P(x)$.

Коэффициент эксцесса

$$E = \frac{1}{ND^2} \sum_{i=1}^N (x_i - \bar{x}) = \frac{M_4}{M_2^2} - 3$$

характеризует степень остроты пика кривой $P(x)$ в сравнении с $P(x)$ для нормального распределения. Если $E > 0$, $P(x)$ имеет более острый пик, чем при нормальном распределении, если $E < 0$ — пик менее острый.

Вспомогательные коэффициенты

$$\alpha_3 = U_3 = \sqrt{\frac{6(N-1)}{(N+1)(N+3)}}$$

$$\alpha_4 = U_4 = \sqrt{\frac{24(N-2)(N-3)}{(N-1)^2(N+3)(N+5)}}$$

служат для приближенной проверки гипотезы о нормальном распределении x_i . Если

$$A < \alpha_3 / (2 \div 3) \text{ и } E < \alpha_4 / (2 \div 3),$$

то распределение $P(x)$ для массива x_i можно считать нормальным.

При программировании вычислений одномерной статистики желательно предусмотреть следующие возможности: 1) накопление сумм x_i^k ; 2) возможность исключения ошибочно введенного числа x_i ; 3) подсчет N в ходе ввода x_i ; 4) выдачу статистических характеристик в любой момент (до окончания ввода всех x_i). Все это обеспечивает приведенная ниже программа.

Программа 5.12.

```

10 PRINT 'СТАТИСТИЧЕСКАЯ ОБРАБОТКА МАССИВА ЧИСЕЛ XI'
20 PRINT 'ИНСТРУКЦИЯ! ДЛЯ ИСКЛЮЧЕНИЯ ОШИБОЧНО ВВЕДЕННОГО'
30 PRINT 'ЧИСЛА НАЖМИТЕ КЛАВИШУ СБР И ДАЙТЕ КОМАНДУ GOTO 80'
40 PRINT 'ДЛЯ ВЫВОДА РЕЗУЛЬТАТОВ НАЖМИТЕ КЛАВИШУ СБР И'
50 PRINT 'ДАЙТЕ КОМАНДУ GOTO 100'
55 LET A=0: LET B=0: LET C=0: LET D=0: LET I=0
60 PRINT!4.0!'ВВЕДИТЕ X' I+1: INPUT X: LET I=I+1
70 LET A=A+X: LET B=B+X*X: LET C=C+X*X*X: LET D=D+X*X*X: GOTO 60
80 LET I=I-1: LET A=A-X: LET B=B-X*X: LET C=C-X*X*X: LET D=D-X*X*X: GOTO 60
90 LET D=D-X*X*X: GOTO 60
100 LET M1=A/I: LET M2=B/I: LET M3=C/I: LET M4=D/I
110 LET D1=M2-M1*M1: LET D2=D1*I/(I-1)
120 LET K=M3-3*M1*M2+2*M1*M1*M1
130 LET W=M1*M1: LET L=M4-4*M1*M3+6*W*M2-3*W*W
140 LET A1=K/D1^1.5: LET E=L/D1/D1-3
150 LET U3=SQR(6*(I-1)/(I+1)/(I+3))
160 LET U4=SQR(24*I*(I-2)*(I-3)/(I+3)/(I+5)/(I-1)^2)
170 PRINT!F1.9!'СРЕДНЕЕ M1=*M1: PRINT 'ДИСПЕРСИЯ D=*D1
180 PRINT 'ДИСПЕРСИЯ D0=*D2: PRINT 'АСИММЕТРИЯ A=*A1
190 PRINT 'ЭКССЕСС E=*E: PRINT 'ЦЕНТРАЛЬНЫЙ МОМЕНТ M3=*K
200 PRINT 'ЦЕНТРАЛЬНЫЙ МОМЕНТ M4=*L: PRINT 'ВВЕДЕНО ЧИСЕЛ N=*!4.0! I
210 PRINT!F1.9!'КОЭФФИЦИЕНТ U3=*U3
220 PRINT 'КОЭФФИЦИЕНТ U4=*U4: GOTO 60: END

```

Коэффициент асимметрии

$$A = \frac{1}{ND^{3/2}} \sum_{i=1}^N (x_i - \bar{x})^3 = M_3 / M_2^{3/2}$$

характеризует скошенность графической функции плотности распределения вероятностей $P(x)$. При $A=0$ она симметрична, при $A > 0$

Пример. Допустим, надо найти статистические параметры массива из 10 чисел x_i : 9, 8, 10, 9, 11, 12, 10, 10, 9 и 11. Пустив программу командой RUN, вводим числа x_i . При этом индицируется номер i . Если какое-то число введено ошибочно, достаточно нажать клавишу СБР и дать команду GOTO 80 (фиксируется нажатием клавиши ПС). По

окончании ввода всех чисел (или их части) нажимается клавиша СБР и дается команда GOTO 100. Для указанных выше чисел получаем: среднее $m_1=9,9$; дисперсия $D=1,29$; дисперсия $D_0=1,433333333$; $A=1,965657837 \cdot 10^{-1}$; $E=-7,539210384 \cdot 10^{-1}$; $M_3=0,288$; $M_4=3,7377$; $N=10$ (число введенных чисел); $U_3=0,6145098678$ и $U_4=0,9224435941$. Сразу после получения результатов можно продолжить ввод последующих x_i .

Среднее геометрическое для N чисел x_i определяется как

$$G_M = (x_1 \cdot x_2 \cdot \dots \cdot x_N)^{1/N}$$

Гармоническое среднее

$$H_M = N / (x_1^{-1} + x_2^{-1} + \dots + x_N^{-1})$$

Программа 5.13.

```

10 PRINT'ВЫЧИСЛЕНИЕ ГЕОМЕТРИЧЕСКОГО СРЕДНЕГО GM'
20 PRINT' И ГАРМОНИЧЕСКОГО СРЕДНЕГО HM'
30 PRINT'ИНСТРУКЦИЯ! ДЛЯ ВВОДА РЕЗУЛЬТАТОВ ЗАДАЙТЕ'
40 PRINT' ПОСЛЕ ВВОДА ПОСЛЕДНЕГО XI НУЛЬ'
50 LETI=0:LETS=0:LETR=1
60 PRINT'ВВЕДИТЕ X'14.0!I+1:INPUT X:IF X=0 GOTO 80
70 LETR=R*X:LETS=S+1/X:LETI=I+1:GOTO 60
80 LETS=R*(1/I):PRINT!F1.9!'СРЕДНЕЕ ГЕОМЕТРИЧЕСКОЕ GM='G
90 LETH=I/S:PRINT'СРЕДНЕЕ ГАРМОНИЧЕСКОЕ HM='H:END

```

Пример. Для вычисления G_M и H_M по этой программе после ввода последнего числа x_N нужно ввести 0. Например, для пяти чисел 2; 3,5; 6,1; 1,2 и 3,9 получим $G_M=2,88492645$ и $H_M=2,45171074$.

Гистограмма распределения характеризует количество чисел x_i , попадающих в интервалы изменения x с границами $d_0, d_1, d_2, \dots, d_N$. Гистограмма распределения графически строится в виде столбцов, высота которых соответствует количеству x_i , попавших в интервал изменения x , на который опирается столбик (на горизонтальной оси x).

Гистограмма интегрального распределения характеризует количество чисел, попадающих в интервалы $(-\infty, d_1), (-\infty, d_2), \dots, (-\infty, d_N)$.

```

10 REM'ПОСТРОЕНИЕ ГИСТОГРАММ МАССИВА ЧИСЕЛ X'
15 PRINT'ДЛЯ ВВОДА РЕЗУЛЬТАТОВ ПОСЛЕ ВВОДА XI НАЧАТЬ'
20 PRINT'КЛАВИШУ СБР И ДАТЬ КОМАНДУ GOTO 100'
30 INPUT'ВВЕДИТЕ X0='X0:INPUT'ВВЕДИТЕ XK='X1
40 INPUT'ВВЕДИТЕ ЧИСЛО ИНТЕРВАЛОВ ГИСТОГРАММЫ N='N
50 DIM A(N):LETI=1:LETC=(X1-X0)/N
70 PRINT!4.0!'ВВЕДИТЕ X('I')=':INPUT X
80 LETI=I+1:LETY=INT(C*(X-X0))
90 LETA(Y)=A(Y)+1:GOTO 70
100 LETS=0:FOR J=0 TO N-1:LETS=S+A(J)
110 PRINT!4.0!'ДЛЯ J='J' NJ='A(J)' SJ='S
120 NEXT J:PRINT'ВВЕДЕНО' I-1' ЧИСЕЛ' GOTO 70:END

```

Алгоритмы подготовки гистограмм. Простейший алгоритм подготовки гистограмм заключается в сравнении x_i с сеткой границ d_1, d_2, \dots, d_N с помощью операций условных переходов и подсчете числа попадающих x_i в каждый интервал. Однако программа при большом числе интервалов получается громоздкой, а главное, большим получается время обработки каждого x_i . В боль-

шинстве случаев отрезки $[d_1, d_2], [d_2, d_3]$ и т. д. имеют одинаковую ширину

$$C = \Delta x = (x_M - x_0) / N, \quad (5.22)$$

где x_M — максимальное значение x_i , x_0 — минимальное значение x_i , N — число интервалов. Тогда алгоритм ускоренной подготовки данных для построения гистограмм будет следующий.

1. Вводим x_M, x_0 и N .

2. Определяем массив счетчиков $A(N)$, вычисляем C по формуле (5.22) и полагаем $I=1$.

3. Организуем ввод x_i , вычисление $I \leftarrow I + 1$ (формирование текущего номера i каждого x_i) и вычисление вспомогательной переменной

$$Y = \text{int}(C \times (x - x_0)).$$

Значение Y указывает на номер того отрезка, в который попадает заданное значение x_i .

4. Вычисляем $A(Y) \leftarrow A(Y) + 1$ (т. е. вносим 1 в содержимое счетчика $A(Y)$) и возвращаемся к п. 3.

5. Вывод данных для построения гистограмм организуем с помощью цикла с управляющей переменной J , меняющейся от значения 1 до N с шагом 1. Числа попадания x_i в J -й отрезок получаем выводом значений $A(J)$. Для получения данных для построения гистограммы интегрального распределения вычисляем $S \leftarrow S + A(J)$ при начальном $S=0$.

Программа 5.14.

Пример. После ввода всех x_i , для вывода данных к построению гистограмм следует нажать клавишу СБР пульта дисплея и дать команду GOTO 100. Для $x_0=5$, $x_M=10$, $N=5$ и $x_i=6,2; 7,1; 9,6; 8,3; 7,4; 8,1; 6,7; 5,5; 7,2; 7,6; 7,9; 9,7; 8,5; 6,8$ и 8,4 получим следующие результаты, которые могут в дальнейшем использоваться для графического представления гистограмм:

J=0 NJ=1 SJ=1
 J=1 NJ=3 SJ=4
 J=2 NJ=5 SJ=9
 J=3 NJ=4 SJ=13
 J=4 NJ=2 SJ=15

Далее, вводя соответствующие операторы (см. выше), получим

MX EXE 137.7142857 (среднее \bar{x})
 F₁ SDX EXE 18,42898069 (σ)
 F₁ SDXN EXE 18,2968716 (σ_0)

Статистические расчеты на специализированных ПЭВМ обычно выполняются микропрограммой. Так, вычисление статистических параметров для $x_i=55, 54, 51, 55, 53, 54$ и 52 на ПЭВМ FX=702P проводится в следующем порядке.

1. Нажав клавиши F₁ и SAC (статистика, общее стирание), обнуляем накапливающие регистры.

2. Вводим x_i набором каждого числа и нажатием клавиши STAT (статистика). Если набранное (но не обработанное) число ошибочно, его исключают нажатием клавиш AC (полный сброс числа) или C (сброс последней цифры). Если ошибочно введенное число обработано (была нажата клавиша STAT), оно исключается нажатием клавиш F₁ и DEL/STAT.

3. Результаты вычислений получаем набором операторов, выполнение которых фиксируется нажатием клавиши EXE:

Описанные операторы могут использоваться и в программах, что упрощает их построение при более сложных статистических расчетах, например, если надо вычислять также коэффициенты асимметрии и эксцесса. Микропрограммная реализация значительной части статистических расчетов существенно повышает скорость статистических вычислений и уменьшает вероятность ошибки.

§ 5.4. Реализация метода Монте-Карло

Метод Монте-Карло заключается в использовании случайных чисел для моделирования различных объектов, ситуаций и физических явлений, реализации игр (подобных игре в карты) и др.

Оператор	Результат	Наименование результата
CNT	8	Количество чисел N
SX	427	Сумма чисел x_i
SX2	22805	Сумма x_i^2
MX	53.375	Среднее \bar{x}
SDX	1.407885953	Стандартное отклонение σ
SDXN	1.316956719	Стандартное отклонение σ_0

Возможна обработка групп чисел, например:

Номер группы	1	2	3	4	5
Число	110	130	150	170	190
Количество чисел в группе	10	31	24	2	3

В этом случае вводятся число, затем знак \square и количество чисел в группе. Например, для ввода и обработки первой группы вводим команды

110 \square 10 \square STAT

С помощью команды F₁ ASTAT/STAT можно (после ввода всех групп) обеспечить выдачу в автоматическом режиме следующих данных (пример для приведенной совокупности чисел):

CNT=70 (общее количество чисел)
 SX=9640 (сумма x_i)
 SY=0 (сумма $y_i=0$)
 SX2=1351000 (сумма x_i^2)
 SY2=0 (сумма $y_i^2=0$)
 SXY=0 (сумма $x_i y_i=0$)

В этом режиме выдаются также параметры массива y_i , элементы которого не вводились (поэтому значения SY, SY2 и SXY нулевые).

Равномерно-распределенные случайные числа обычно генерируются ЭВМ в отрезке значений [0; 1], причем любое значение V_i в этом интервале равновероятно. Обычно для этого используется отделение дробной части от сложного арифметического выражения, содержащего предшествующее число V_i . Например, может использоваться соотношение вида

$$V_{i+1} = \text{FRAC}(kV_i),$$

где $k=8t \pm 3$ и t — нечетное целое число (при $t=5$ $k=37$ или $k=43$). Обычно перед использованием датчика случайных чисел задается начальное значение V_0 на отрезке [0; +1]. Задание разных V_0 позволяет формировать различные последовательности случайных чисел. В действительности получаются «квазислучайные» числа, т. е. спустя некоторое количество циклов последовательности чисел повторяются. Количество неповторяющихся чисел находится в пределах от нескольких тысяч до сотен тысяч.

Перевод равномерно-распределенных случайных чисел на отрезке $[a, b]$ производится с помощью формулы

$$X_{i+1} = a + (b-a) V_{i+1}.$$

Случайные числа с различными законами распределения получают с помощью формул преобразования (см. [10, 38]).

Случайные числа с нормальным распределением могут быть получены с помощью формул

$$R'_i = \sqrt{2 \ln(1/V_i)} \cos(2\pi V_{i-1}),$$

$$R''_i = \sqrt{2 \ln(1/V_i)} \sin(2\pi V_{i-1}).$$

При этом получается сопряженная пара чисел, имеющих среднее значение $\bar{R}=0$ и среднеквадратичное отклонение $\sigma=1$. При других значениях \bar{R} и σ производится пересчет по формуле

$$r_{i+1} = \bar{R} + R_{i+1}\sigma.$$

Обобщенный алгоритм реализации метода Монте-Карло (рис. 5.3) обеспечивает моделирование работы объекта и вычисление основных статистических характеристик его функциональных параметров.

Наглядным примером применения метода Монте-Карло является программа 5.13 для обучения школьников младших классов таблице умножения. Программа генерирует два случайных числа, переводит их в отрезок [1; 9] и предлагает учащемуся перемножить эти числа. Результат сравнивается с правильным. Оценка выставляется по результатам анализа десяти операций перемножения. Читателю предлагается самостоятельно разобрать эту программу.

Программа 5.15.

```

05 PRINT 'ОБУЧЕНИЕ ТАБЛИЦЕ УМНОЖЕНИЯ'
10 LET X=.7654321:LET N=0
20 FOR I=1 TO 10
30 LET A=RND(X):LET B=RND(X)
40 LET A=INT(A*9+1):LET B=INT(B*9+1)
50 PRINT!2.0!'СКОЛЬКО БУДЕТ 'A'*B'=?': INPUT'C
60 LET D=A*B: IF C=D THEN 80
70 PRINT 'НЕВЕРНО' A'*B'='D':LET N=N+1:GOTO 85
80 PRINT 'ПРАВИЛЬНО'
85 NEXT I
90 IF N=0 THEN 140
100 IF N<=1 THEN 150
110 IF N<=3 THEN 160
120 PRINT 'ВЫ НЕ ЗНАЕТЕ ТАБЛИЦУ УМНОЖЕНИЯ'
130 PRINT 'ПОУПРАЖНЯЙТЕСЬ ЕЩЕ':GOTO 10
140 PRINT 'ОТЛИЧНО.ВСЕ РЕШЕНО ВЕРНО':GOTO 10
150 PRINT 'ХОРОШО.ОДНА ОШИБКА':GOTO 10
160 PRINT 'ПОСРЕДСТВЕННО.ДОПУЩЕНЫ 'N' ОШИБКИ':GOTO 10
170 END
    
```

Отметим основные области применения метода Монте-Карло: решение нелинейных уравнений, вычисление экстремумов (в том числе глобальных у многоэкстремальных функций многих переменных), вычисление определенных интегралов большой кратности и др.

§ 5.5. Корреляционный анализ

Корреляция является признаком, указывающим на взаимосвязь ряда численных последовательностей. Парная корреляция характеризует взаимосвязь двух последовательностей x_i и y_i .

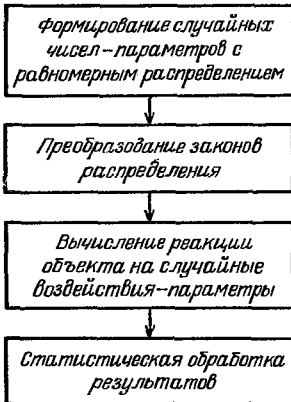


Рис. 5.3. Обобщенный алгоритм применения метода Монте-Карло

Корреляционный момент двух последовательностей

$$R_{x,y} = \frac{1}{N} \sum_{i=1}^N (y_i - m_{1y}) (x_i - m_{1x}).$$

Коэффициент парной корреляции

$$R = \frac{\sum_{i=1}^N x_i y_i - \left(\sum_{i=1}^N x_i \sum_{i=1}^N y_i \right) / N}{\sqrt{\sum_{i=1}^N x_i^2 - \frac{\left(\sum_{i=1}^N x_i \right)^2}{N}}} \sqrt{\sum_{i=1}^N y_i^2 - \frac{\left(\sum_{i=1}^N y_i \right)^2}{N}} \quad (5.23)$$

характеризует степень отклонения связи между x_i и y_i от линейной. Если $|R|$ близок к 1, то эта связь линейна, т. е. $y_i = ax_i + b$,

причем знак R определяется знаком коэффициента a . Если $R > 0$, то $a > 0$, и напротив, при $R < 0$ $a < 0$.

Программа 5.16.

```
05 PRINT 'ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТА ПАРНОЙ КОРРЕЛЯЦИИ'
10 INPUT 'ВВЕДИТЕ N=' N
20 LET A=0: LET B=0: LET C=0: LET D=0: LET E=0
30 PRINT 'ВВЕДИТЕ ПОПАРНО X(I), Y(I)'
40 FOR I=1 TO N: PRINT I: I='I
45 INPUT 'X(I)=' X: INPUT 'Y(I)=' Y
50 LET A=A+X: LET B=B+Y: LET C=C+X^2
60 LET D=D+Y^2: LET E=E+X*Y: NEXT I
70 LET R=(E-A*B/N)/SQRT((C-(A^2)/N)*(D-(B^2)/N))
80 PRINT 'F1.9!' 'КОЭФ. КОРРЕЛЯЦИИ R=' R
```

Пример. Для $N=5$ и $y_i(x_i)$, равных $2(0,95)$; $4,05(2,1)$; $5,8(3)$; $8,1(4,1)$ и $9,2(4,9)$, получим $R=0,9987280191$.

Порядковая корреляция по Спирмену оценивается коэффициентом порядковой корреляции [18]

$$R_n = 1 - \frac{6}{N(N^2-1)} \sum_{k=1}^N (A_k - B_k)^2,$$

где A_k и B_k — место (ранг), которое занимают x_k и y_k при убывании x_k .

Программа 5.17.

```
10 PRINT 'ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТА РАНГОВОЙ КОРРЕЛЯЦИИ'
20 INPUT 'ВВЕДИТЕ N=' N: LET S=0
30 FOR I=1 TO N: PRINT I: I='I
40 INPUT 'ВВЕДИТЕ A, B ' A, B
50 LET S=S+(A-B)^2: NEXT I
60 LET R=1-6*S/N/(N^2-1)
70 PRINT 'F1.9!' 'КОЭФФИЦИЕНТ R КОП=' R: END
```

Пример. Для $N=7$ и $A(B)$, равных $5(6)$, $4(7)$, $3(3)$, $7(5)$, $2(1)$, $1(2)$, $6(4)$, получим $R_n=0,6428571429$.

Вычисление коэффициента корреляции на специализированных ПЭВМ выполняется микропрограммно. Так, на ПЭВМ FX-702P для этого выполняются следующие операции.

1. Нажатием клавиш F_1 и SAC обнуляются ячейки памяти статистических расчетов.

2. По схеме

DEL

x_i y_i [STAT]

вводятся пары чисел x_i и y_i . Ошибочно введенная и обработанная пара исключается нажатием клавиш F_1 и DEL/STAT.

3. Вводом команды COR вычисляется R .

§ 5.6. Регрессионный анализ (приближение функций по методу наименьших квадратов)

Линейный парный регрессионный анализ заключается в определении параметров эмпирической линейной зависимости

$$y(x) = b_1x + b_0, \quad (5.24)$$

описывающей связь между некоторым числом N пар значений x_i и y_i , обеспечивая при этом наименьшую среднеквадратичную погрешность. Графически эту задачу можно

представить следующим образом — в облаке точек $x_i y_i$ плоскости $x y$ (рис. 5.4) требуется провести прямую так, чтобы величина всех отклонений отвечала условию

$$U = \sum_{i=1}^N [y_i - y(x_i)]^2 = \min,$$

где $y(x_i)$ — зависимость (5.24). Для этого

нужно приравнять нулю частные производные

$$\frac{\partial U}{\partial b_0} = \sum_{i=1}^N [y_i - (b_0 + b_1x_i)],$$

$$\frac{\partial U}{\partial b_1} = \sum_{i=1}^N [y_i - (b_0 + b_1x_i)] x_i,$$

что дает для определения неизвестных коэффициентов b_0 и b_1 систему линейных

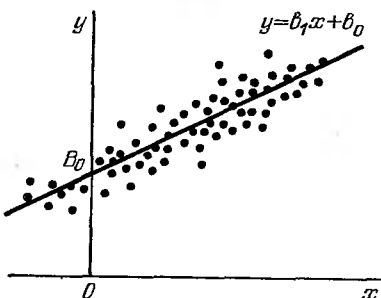


Рис. 5.4. К линейной регрессии

уравнений

$$b_0 N + b_1 \sum_{i=1}^N x_i = \sum_{i=1}^N y_i,$$

$$b_0 \sum_{i=1}^N x_i + b_1 \sum_{i=1}^N x_i^2 = \sum_{i=1}^N x_i y_i.$$

Решение этой системы:

$$b_1 = \frac{\sum_{i=1}^N x_i \sum_{i=1}^N y_i - N \sum_{i=1}^N x_i y_i}{\left(\sum_{i=1}^N x_i\right)^2 - N \sum_{i=1}^N x_i^2},$$

$$b_0 = \frac{1}{N} \left(\sum_{i=1}^N y_i - b_1 \sum_{i=1}^N x_i \right).$$

Программа 5.18.

```

05 PRINT 'ЛИНЕЙНАЯ РЕГРЕССИЯ'
10 INPUT 'ВВЕДИТЕ N=' N
20 LET A=0: LET B=0: LET C=0: LET D=0
30 PRINT '3.0! ВВЕДИТЕ X(I), Y(I)'
40 FOR I=1 TO N: PRINT 'I=' I
50 INPUT 'X(I)=' X: INPUT 'Y(I)=' Y
60 LET A=A+X: LET B=B+Y: LET C=C+X^2
70 LET D=D+X*Y: NEXT I
80 LET B1=(A*B-N*D)/(A^2-N*C)
90 LET B0=(B-B1*A)/N
100 PRINT 'F1.9! Y=' B0 + 'B1'*X'
110 INPUT 'ВВЕДИТЕ X=' X
120 PRINT 'Y(X)=' B0+B1*X: GOTO 110: END
    
```

Пример. Пусть имеется совокупность из $N=5$ пар $y_i(x_i)$: 5,5(2); 6,3(4); 7,2(6); 8(8); 8,6(10). Введя эти данные, получим $b_0=4,75$ и $b_1=0,395$. Следовательно, зависимость $y(x)$ имеет вид

$$y(x) = 4,75 + 0,395x.$$

Программа предусматривает вычисление y по заданному значению x .

При необходимости программа может быть легко дополнена вычислением коэффициента парной корреляции R (5.23) или среднеквадратичной погрешности

$$\sigma^2 = \frac{1}{N} \left[\sum_{i=1}^N y_i^2 - b_1 \sum_{i=1}^N y_i - b_0 \sum_{i=1}^N x_i y_i \right],$$

что позволит количественно оценивать степень приближения точек $x_i y_i$ к прямой (5.24).

Линейная регрессия с помощью специализированных ПЭВМ выполняется микропрограммно. Так, у ПЭВМ FX-702P предусмотрены операторы LRA и LRB для вычисления параметров $A=b_0$ и $B=b_1$ линейной регрессии $y(x)=A+Bx$, а также операторы EOY для вычисления y по заданному x и EOX для вычисления x по заданному y . Одновременно при этом вычисляется коэффициент корреляции R и другие статистические параметры массивов x_i и y_i , описанные ранее (всего 17 параметров).

Последовательность вычислений рассмотрим на примере к программе 5.18 (см. ниже).

Как показывает данный пример, ПЭВМ FX-702P, специализированные на научно-технические и статистические расчеты, обеспечивают микропрограммно (т. е. без ввода внешних программ) вычисление важнейших характеристик двумерных массивов $x_i y_i$, а также проведение корреляционного и линейного регрессионного анализа. Указанные выше операторы могут использоваться и в составе внешних программ, что резко упрощает последние.

Нелинейная парная регрессия сводится к получению заданной нелинейной зависимости $y(x)$ (нелинейной по независимой переменной x , но линейной по параметрам этой зависимости!), приближающей совокупность

Обнуление регистров памяти

Ввод 1 пары x_i, y_i

Ввод 2 пары x_i, y_i

Ввод 3 пары x_i, y_i

Ввод 4 пары x_i, y_i

Ввод 5 пары x_i, y_i

Вычисление A

Вычисление B

Вычисление σ_x

Вычисление σ_{0x}

Вычисление σ_y

Вычисление σ_{0y}

Вычисление R

Вычисление N

Вычисление суммы x_i

Вычисление суммы y_i

Вычисление суммы x_i^2

Вычисление суммы y_i^2

Вычисление суммы $x_i y_i$

Вычисление \bar{x}

Вычисление \bar{y}

Вычисление y при $x=5$

Вычисление x при $y=6,725$

F1	SAC	
2, 5.5	STAT	
4, 6.3	STAT	
6, 7.2	STAT	
8, 8	STAT	
10, 8.6	STAT	
F1 LRA	EXE	4.75
F1 LRB	EXE	0.395
F1 SDX	EXE	3.16227766
F1 SDXN	EXE	2.828427125
F1 SDY	EXE	1.251798706
F1 SDYN	EXE	1.1196428
F1 COR	EXE	0.9978438785
F1 ASTAT	CNT=5	
	SX=30	
	SY=35.6	
	SX2=220	
	SY2=259.74	
	SXY=229.4	
	MX	EXE 6
	MY	EXE 7.12
	EOY 5	EXE 6.725
	EOX 6.725	EXE 5

чисел x_i и y_i с наименьшей среднеквадратической погрешностью.

Сведение нелинейной регрессии к линейной выполняется с помощью линеаризирующих преобразований в ходе ввода x_i , y_i и при выводе b_0 и b_1 (см. табл. 5.1). Такое сведение целесообразно, если ПЭВМ имеет микропрограммное вычисление параметров b_0 и b_1 линейной регрессии или в распоряжении пользователя имеется только такая программа.

Пример. Зависимость $y_i(x_i)$, заданная числами 21,4(6,9); 15,7(12,9); 12,1(19,8); 8,5(26,7) и 5,2(35,1) близка к экспоненциальной $y = Ae^{bx}$. Для нахождения A и B ввод x_i и y_i проводим по схеме (для ПЭВМ FX-702P)

x_i LN y_i STAT

т. е. вместо y_i вводим $\ln y_i$. Тогда константу $A = 0,2388010829$ получим, введя команды EXP LRA EXE, а константу $B = 2,771866148$ получим, введя команды LRB EXE.

В общем случае для нелинейной регрессии более удобно использовать специальные программы, не требующие специальных преобразований при вводе x_i , y_i , а также при выводе результатов вычислений.

Гиперболическая регрессия заключается в нахождении параметров функции

$$y(x) = b_0 + b_1/x$$

из решения системы уравнений

$$b_0 N + b_1 \sum_{i=1}^N 1/x_i = \sum_{i=1}^N y_i$$

$$b_0 \sum_{i=1}^N 1/x_i + b_1 \sum_{i=1}^N 1/x_i^2 = \sum_{i=1}^N y_i/x_i$$

Программа 5.1.

```

05 PRINT 'ГИПЕРБОЛИЧЕСКАЯ РЕГРЕССИЯ'
10 INPUT 'ВВЕДИТЕ N=' N
20 LET A=0: LET B=0: LET C=0: LET D=0
30 PRINT 'ВВЕДИТЕ ПОПАРНО X(I), Y(I) '
40 FOR I=1 TO N: PRINT!3.0!' I=' I
50 INPUT 'X(I)=' X: INPUT 'Y(I)=' Y
60 LET A=A+1/X: LET B=B+1/(X^2)
70 LET C=C+Y: LET D=D+Y/X:NEXT I
80 LET E=N*B-A^2: LET F=C*B-D*A
90 LET K=N*D-A*C: LET F=F/E: LET K=K/E
100 PRINT!F1.9!' Y(X)=' F+' K'/X'
110 INPUT 'ВВЕДИТЕ X=' X
120 PRINT 'Y(X)=' F+K/X:GOTO 110:END
    
```

Пример. Для $N=8$ и $y_i(x_i)$, заданных значениями 12,2(1); 6,8(2); 5,2(3); 4,6(4); 3,9(5); 3,7(6); 3,5(7) и 3,2(8), получим $b_0 = 1,935761896$ и $b_1 = 10,16017523$. При $x=2$ получим $y = 7,015849511$.

Степенная регрессия обеспечивает нахождение параметров функции

$$y = b_0 x^{b_1}$$

по формулам

$$b_1 = \frac{\sum_{i=1}^N \ln x_i \sum_{i=1}^N \ln y_i - N \sum_{i=1}^N \ln x_i \ln y_i}{\left(\sum_{i=1}^N \ln x_i \right)^2 - N \sum_{i=1}^N (\ln x_i)^2}$$

$$b_0 = \exp \left[\frac{1}{N} \left(\sum_{i=1}^N \ln y_i - b_1 \sum_{i=1}^N \ln x_i \right) \right]$$

Таблица 5.1

Преобразования, сводящие нелинейную регрессию к линейной
(ее параметры помечены штрихами)

№	Функция $y(x)$	x'	y'	b_0	b_1
1	$b_0 + b_1 x$	x	y	b_0'	b_1'
2	$1/(b_0 + b_1 x)$	x	$1/y$	b_0'	b_1'
3	$b_0 + b_1/x$	$1/x$	y	b_0'	b_1'
4	$x/(b_0 + b_1 x)$	x	x/y	b_0'	b_0'
5	$b_0 b_1^x$	x	$\lg y$	$10 b_0'$	$10 b_1'$
6	$b_0 \exp(b_1/x)$	x	$\ln y$	$\exp b_0'$	b_1'
7	$b_0 10^{b_1 x}$	x	$\lg y$	$10^{b_0'}$	b_1'
8	$1/(b_0 + b_1 e^{-x})$	e^{-x}	$1/y$	b_0'	b_1'
9	$b_0 x^{b_1 x}$	$\lg x$	$\lg y$	$10^{b_0'}$	b_1'
10	$b_0 + b_1 \lg x$	$\lg x$	y	b_0'	b_1'
11	$b_0 + b_1 \ln x$	$\ln x$	y	b_0'	b_1'
12	$b_0/(b_1 + x)$	x	$1/y$	$1/b_1'$	b_0'/b_1'
13	$b_0 x/(b_1 + x)$	$1/x$	$1/y$	$1/b_1'$	b_0'/b_1'
14	$b_0 \exp(b_1/x)$	$1/x$	$\ln x$	$\exp b_0'$	b_1'
15	$b_0 10^{b_1/x}$	$1/x$	$\lg x$	$10^{b_0'}$	b_1'
16	$b_0 + b_1 x^n$	x^n	y	b_0'	b_1'

Программа 5.20.

```

05 PRINT'СТЕПЕННАЯ РЕГРЕССИЯ'
10 INPUT'ВВЕДИТЕ N='N
20 LETA=0:LETB=0:LETC=0:LETD=0
30 PRINT'ВВЕДИТЕ ПОПАРНО X(I),Y(I)'
40 FOR I=1 TO N:PRINT!3.0!'I='I
50 INPUT'X(I)='X:INPUT'Y(I)='Y
60 LETA=A+LOG(X):LETB=B+LOG(Y)
70 LETC=C+LOG(X)*LOG(Y):LETD=D+LOG(X)^2:NEXT I
80 LETE=(A*B-N*C)/(A^2-N*D)
90 LETF=EXP((B-E*A)/N)
100 PRINT!F1.9!'Y(X)='F'*X^E
110 INPUT'ВВЕДИТЕ X='X
120 PRINT'Y(X)='F*X^E:GOTO 110:END
    
```

Пример. Для $N=6$ и $y_i(x_i)$, заданных числами 3(1), 12(2), 27(3), 48(4), 75(5) и 108(6), получим $b_0=2,9999999987$ и $b_1=2,000000004$. Для $x=2$ получим $y=11,99999998$.

Показательная регрессия обеспечивает получение параметров a и b показательной функции

$$y(x) = ab^x$$

из решения системы уравнений

$$N \lg a + \lg b \sum_{i=1}^N x_i = \sum_{i=1}^N \lg y_i,$$

$$\lg a \sum_{i=1}^N x_i + \lg b \sum_{i=1}^N x_i^2 = \sum_{i=1}^N (x_i \lg y_i).$$

Экспоненциальная регрессия обеспечивает получение параметров функции

$$y = b_0 \exp(b_1 x)$$

по формулам

$$b_1 = \frac{\sum_{i=1}^N x_i \sum_{i=1}^N \ln y_i - N \sum_{i=1}^N x_i \ln y_i}{\left(\sum_{i=1}^N x_i\right)^2 - N \sum_{i=1}^N x_i^2},$$

$$b_0 = \exp \left[\frac{1}{N} \left(\sum_{i=1}^N \ln y_i - b_1 \sum_{i=1}^N x_i \right) \right].$$

Программа 5.21.

```

05 PRINT'ПОКАЗАТЕЛЬНАЯ РЕГРЕССИЯ'
10 INPUT'ВВЕДИТЕ N='N
20 LETA=0:LETB=0:LETC=0:LETD=0
30 PRINT'ВВЕДИТЕ ПОПАРНО X(I),Y(I)'
40 FOR I=1 TO N:PRINT!3.0!'I='I
50 INPUT'X(I)='X:INPUT'Y(I)='Y
60 LETA=A+X:LETB=B+X^2:LETC=C+LGT(Y)
70 LETD=D+X*LG(Y):NEXT I
80 LETE=N*B-A^2:LETF=C*B-D*A
90 LETK=N*D-A*B:LETK=EXT(F/E):LETK=EXT(K/E)
100 PRINT!F1.9!'Y(X)='F'*K^X
110 INPUT'ВВЕДИТЕ X='X
120 PRINT'Y(X)='F*K^X:GOTO 110:END
    
```

Пример. Для $N=5$ и $y_i(x_i)$, равных 6(1), 7(2), 8,7(3), 10,4(4) и 12,4(5), получим $a=4,941990005$ и $b=1,202950112$. Для $x=2$, $y=7,151499236$.

Программа 5.22.

```

05 PRINT'ЭКСПОНЕНЦИАЛЬНАЯ РЕГРЕССИЯ'
10 INPUT'ВВЕДИТЕ N='N
20 LETA=0:LETB=0:LETC=0:LETD=0
30 PRINT'ВВЕДИТЕ ПОПАРНО X(I),Y(I)'
40 FOR I=1 TO N:PRINT!3.0!'I='I
50 INPUT'X(I)='X:INPUT'Y(I)='Y
60 LETA=A+X:LETB=B+LOG(Y):LETC=C+X^2
70 LETD=D+X*LOG(Y):NEXT I
80 LETB1=(A*B-N*D)/(A^2-N*C)
90 LETB0=EXP((B-B1*A)/N)
100 PRINT!F1.9!'Y(X)='B0'*EXP('B1'*X)
110 INPUT'ВВЕДИТЕ X='X
120 PRINT'Y(X)='B0*EXP(B1*X):GOTO 110:END
    
```

К программе 5.22.

Пример. Для $N=9$ и $y_i(x_i)$, заданных значениями 3,5(2), 5(3), 6,2(4), 9(5), 13(6), 16(7), 23(8), 30(9) и 40(10), получим $b_0=$

$= 1,939481364$ и $b_1 = 0,305283305$. При $x = 5$ получим $y = 8,924828629$.

Логарифмическая регрессия дает параметры a и b функции

$$y = a + b \lg x$$

из решения системы уравнений

$$aN + b \sum_{i=1}^N \lg x_i = \sum_{i=1}^N y_i,$$

$$a \sum_{i=1}^N \lg x_i + b \sum_{i=1}^N (\lg x_i)^2 = \sum_{i=1}^N (y_i \lg x_i).$$

$$b_0 N + b_1 \sum_{i=1}^N x_i + b_2 \sum_{i=1}^N x_i^2 = \sum_{i=1}^N y_i,$$

$$b_0 \sum_{i=1}^N x_i + b_1 \sum_{i=1}^N x_i^2 + b_2 \sum_{i=1}^N x_i^3 = \sum_{i=1}^N x_i y_i,$$

$$b_0 \sum_{i=1}^N x_i^2 + b_1 \sum_{i=1}^N x_i^3 + b_2 \sum_{i=1}^N x_i^4 = \sum_{i=1}^N x_i^2 y_i.$$

Программа 5.23.

```

05 PRINT 'ЛОГАРИФМИЧЕСКАЯ РЕГРЕССИЯ'
10 INPUT 'ВВЕДИТЕ N=':N
20 LET A=0:LET B=0:LET C=0:LET D=0
30 PRINT 'ВВЕДИТЕ ПОПАРНО X(I),Y(I)'
40 FOR I=1 TO N:PRINT!3.0!'I='I
50 INPUT 'X(I)=':X: INPUT 'Y(I)=':Y
60 LET A=A+LGT(X):LET B=B+LGT(X)^2:LET C=C+Y
70 LET D=D+Y*LGT(X):NEXT I
80 LET E=N*B-A^2:LET F=C*B-D*A
90 LET K=N*D-A*C:LET P=F/E:LET K=K/E
100 PRINT!F1.9!'Y(X)='F+'K'*LGT(X)'
110 INPUT 'ВВЕДИТЕ X=':X
120 PRINT 'Y(X)=':F+K*LGT(X):GOTO 110:END

```

Пример. Для $N=6$ и $y_i(x_i)$, равных $1(1); 1,451(2); 1,716(3); 1,903(4); 2,048(5)$ и $2,167(6)$ получим $a = 0,999931767$; $b = 1,499793742$ и для $x=4$ получим $y = 1,902897575$.

Параболическая регрессия обеспечивает получение трех параметров b_0, b_1 и b_2 приближения параболической функцией

$$y(t) = b_0 + b_1 x + b_2 x^2$$

Пример (к программе 5.24). Для $N=7$ и $y_i(x_i)$, заданных значениями $3,76(2); 4,44(4); 5,04(6); 5,56(8); 6(10); 6,36(12)$ и $6,64(14)$, получим $b_0=3, b_1=0,4$ и $b_2=-0,01$. Для $x=7$ получим $y=5,31$.

Полиномиальная регрессия (аппроксимация) обеспечивает нахождение коэффициентов полинома

$$y(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m \quad (5.25)$$

Программа 5.24.

```

05 PRINT 'ПАРАБОЛИЧЕСКАЯ РЕГРЕССИЯ'
10 INPUT 'ВВЕДИТЕ N=':N: LET A=N
20 DATA 0,0,0,0,0,0,0
30 READ B,C,F,M,P,R,S
40 PRINT 'ВВЕДИТЕ ПОПАРНО X(I),Y(I)'
50 FOR I=1 TO N:PRINT!3.0!'I='I
60 INPUT 'X(I)=':X: INPUT 'Y(I)=':Y
70 LET B=B+X: LET C=C+X^2: LET F=F+X^3
80 LET M=M+X^4: LET P=P+Y: LET R=R+X*Y
90 LET S=S+Y*X^2:NEXT I
100 LET D=B:LETE=C:LET K=C:LETL=F
110 LET Q=D/A:LET E=E-Q*B:LET F=F-Q*C
120 LET R=R-Q*P:LET Q=K/A:LETL=L-Q*B
130 LET M=M-Q*C:LET S=S-Q*P:LET Q=L/E
140 LET B2=(S-R*Q)/(M-F*Q): LET B1=(R-F*B2)/E
150 LET B0=(P-B*B1-C*B2)/A:PRINT!F1.9!'B0='B0
160 PRINT 'B1=':B1:PRINT 'B2=':B2
170 INPUT 'ВВЕДИТЕ X=':X
180 PRINT 'Y(X)=':B0+B1*X+B2*X^2:END

```

из решения системы уравнений

$$c_0a_0 + c_1a_1 + c_2a_2 + \dots + c_ma_m = d_0,$$

$$c_1a_0 + c_2a_1 + c_3a_2 + \dots + c_{m+1}a_m = d_1,$$

$$c_ma_0 + c_{m+1}a_1 + c_{m+2}a_2 + \dots + c_{2m}a_m = d_m,$$

где

$$c_j = \sum_{i=1}^N x_i^j, \quad j=0, 1, 2, \dots, 2m, \quad (5.26)$$

$$d_k = \sum_{i=1}^N x_i^k y_i, \quad k=0, 1, 2, \dots, m. \quad (5.27)$$

Полином (5.25) степени $m < N$, где N — число пар x_i и y_i , обеспечивает аппроксимацию (и интерполяцию) таблично заданной функции $y_i(x_i)$ с минимальной среднеквадратичной погрешностью

$$E = \sqrt{\sum_{i=1}^N \epsilon_i^2 / (N+1)}. \quad (5.28)$$

Если $m = N$, то имеет место обычная интерполяция, т. е. значения $y(x)$ (5.25) при $x = x_i$ точно совпадают с заданными y_i . При $m < N$ такого совпадения в общем случае нет. Таким образом, аппроксимация $y_i(x_i)$ по методу наименьших квадратов имеет более универсальный характер, чем обычная интерполяция (§ 4.2).

Если достаточно ограничиться вычислением лишь коэффициентов a_0, a_1, \dots, a_m полинома (5.25) без вычисления значения E , то нет необходимости в хранении массивов x_i и y_i . При этом достаточно вычислить c_j

(массив из $2m$ чисел) и d_k (массив из m чисел) с помощью (5.26) и (5.27) и решить полученную систему линейных уравнений методом Гаусса (см. § 4.1). Таким образом, составлена программа 5.25. Помимо вычисления коэффициентов a_0, a_1, \dots, a_m она обеспечивает нахождение $y(x)$ по заданному x с помощью полученного полинома (5.25).

Программа 5.25.

Пример. Используем пример к программе 5.24 (задаем $m=M=2$ и $n=N=7$). Получим следующий результат: $a_0=A(0)=-3$, $a_1=A(1)=0,4$ и $b_2=A(2)=-0,01$. Задав далее $x=7$, получаем $y(x)=Y(X)=-5,31$.

Время счета контрольного примера (без учета времени ввода x_i и y_i) около 7 с.

Полиномиальная регрессия (аппроксимация) с автоматическим выбором степени полинома выполняется по следующему алгоритму. Вначале задается степень $m=1$ (линейная регрессия). Отсчеты x_i, y_i вводятся и запоминаются. После нахождения всех a_2 (вначале a_0 и a_1) с помощью (5.28) вычисляется среднеквадратичная погрешность E и сравнивается с заданной E_1 . Если $E > E_1$, степень полинома m увеличивается на 1, и т. д. Счет прекращается, как только достигается $E < E_1$. В программе 5.26 наряду с этим алгоритмом возможна реализация алгоритма вычисления коэффициентов a_0, a_1, \dots, a_m полинома заданной степенью m с вычислением получаемой при этом погрешности E . Обеспечивается также вычисление $y(x)$ по заданному x . Для повышения точности система уравнений для вычисления a_0, a_1, \dots, a_m решается методом Гаусса с выбором главного элемента.

Программа 5.25.

```

05 PRINT 'АППРОКСИМАЦИЯ ПОЛИНОМОМ ПО МЕТОДУ НАИМЕНЬШИХ КВАДРАТОВ'
10 INPUT 'ВВЕДИТЕ СТЕПЕНЬ ПОЛИНОМА M=':N:LETN=N+1
20 DIM A(N,N),B(N),C(2*N),X(N)
30 INPUT 'ВВЕДИТЕ ЧИСЛО ПАР X,Y N=':N
40 FOR I=1 TO N:PRINT!3.0!'ДЛЯ I=':INPUT 'ВВЕДИТЕ X,Y ':X,Y
45 LETF=1:FOR J=1 TO 2*N-1:IF J>N THEN 55
50 LETB(J)=B(J)+Y:LETY=Y*X
55 LETC(J)=C(J)+F:LETF=F*X:NEXT J:NEXT I
60 FOR I=1 TO N:LETK=I:FOR J=1 TO N
65 LETA(I,J)=C(K):LETK=K+1:NEXT J:NEXT I
70 FOR I=1 TO N-1:FOR J=I+1 TO N
75 LETA(J,I)=-A(J,I)/A(I,I):FOR K=I+1 TO N
80 LETA(J,K)=A(J,K)+A(J,I)*A(I,K):NEXT K
90 LETB(J)=B(J)+A(J,I)*B(I):NEXT J:NEXT I
100 LETX(N)=B(N)/A(N,N)
110 FOR I=N-1 TO 1 STEP -1:LETH=B(I)
120 FOR J=I+1 TO N:LETH=H-X(J)*A(I,J):NEXT J
130 LETX(I)=H/A(I,I):NEXT I
140 PRINT 'КОЭФФИЦИЕНТЫ ПОЛИНОМА'
150 FOR I=0 TO N-1:PRINT!3.0!'A('I')='!F1.9!X(I+1):NEXT I
160 INPUT 'ВВЕДИТЕ X=':Z:LETS=0:FOR I=N TO 2 STEP -1
170 LETS=(S+X(I))*Z:NEXT I
180 PRINT 'ЗНАЧЕНИЕ Y(X)='!F1.9!S+X(1):GOTO 160:END

```

```

10 PRINT 'АППРОКСИМАЦИЯ ПОЛИНОМОМ ПО МЕТОДУ НАИМЕНЬШИХ КВАДРАТОВ'
15 PRINT 'С ВОЗМОЖНОСТЬЮ АВТОМАТИЧЕСКОГО ВЫБОРА СТЕПЕНИ ПОЛИНОМА'
20 INPUT 'ЗАДАЙТЕ ЧИСЛО УЗЛОВ ИНТЕРПОЛЯЦИИ N=' N
25 DIM X(N), Y(N), Z(N), B(N), G(N), A(N, N), C(N, N), D(2*N)
30 FOR I=1 TO N: PRINT!2.0!'ВВЕДИТЕ X'I', Y'I
35 INPUT X(I), Y(I): NEXT I: LET N=N-1
40 PRINT 'УКАЖИТЕ 0-ЕСЛИ ЗАДАНА ПОГРЕШНОСТЬ ИНТЕРПОЛЯЦИИ'
50 INPUT ' 1-СТЕПЕНЬ ПОЛИНОМА ' U: IF U=0 GOTO 80
60 INPUT 'ВВЕДИТЕ СТЕПЕНЬ ПОЛИНОМА M<N M=' N: GOTO 85
80 INPUT 'ЗАДАЙТЕ СРЕДНЕКВАДРАТИЧНУЮ ПОГРЕШНОСТЬ E1=' E1
85 LET N=N+1: FOR J=0 TO 2*N-1: IF J>N GOTO 95
90 LET B(J)=0
95 LET D(J)=0: NEXT J: FOR I=1 TO N: LET R=Y(I): LET F=1
105 FOR J=1 TO 2*N-1: IF J>N GOTO 115
110 LET B(J)=B(J)+R: LET R=R*X(I)
115 LET D(J)=D(J)+F: LET F=F*X(I): NEXT J: NEXT I
120 FOR I=1 TO N: LET K=I: FOR J=1 TO N
130 LET A(I, J)=D(K): LET K=K+1: NEXT J: NEXT I
140 LET N1=N-1: FOR K=1 TO N1
150 IF ABS(A(K, K))>0 GOTO 230
160 LET K1=K+1: FOR M=K1 TO N: IF ABS(A(M, K))=0 GOTO 210
190 FOR L=1 TO N: LET U=A(K, L): LET A(K, L)=A(M, L)
200 LET A(M, L)=U: NEXT L
210 NEXT M
220 LET U=B(K): LET B(K)=B(M): LET B(M)=U
230 LET G(K)=B(K)/A(K, K): LET K1=K+1
240 FOR I=K1 TO N: LET B(I)=B(I)-A(I, K)*G(K)
250 FOR J1=K TO N: LET J=N-J1+K: LET C(K, J)=A(K, J)/A(K, K)
260 LET A(I, J)=A(I, J)-A(I, K)*C(K, J): NEXT J1: NEXT I: NEXT K
280 LET M=N: LET Z(M)=B(M)/A(M, M)
290 LET M=M-1: LET S=0: FOR L=M TO N1
300 LET S=S+C(M, L+1)*Z(L+1): NEXT L
310 LET Z(M)=G(M)-S: IF M>1 GOTO 290
320 LET E=0: FOR I=1 TO N: LET S=Y(I): LET R=1
330 FOR J=1 TO N: LET S=S-R*Z(J): LET R=R*X(I): NEXT J
340 LET E=E+S*S: NEXT I: LET E=SQR(E/N): IF U=1 GOTO 370
350 IF E<=E1 GOTO 370
360 IF N<N GOTO 85
370 PRINT 'СТЕПЕНЬ ПОЛИНОМА M=' N-1
380 PRINT!F1.9!' ПОГРЕШНОСТЬ ИНТЕРПОЛЯЦИИ E=' E: IF U=1 GOTO 390
385 PRINT 'ЕСЛИ E БОЛЬШЕ ЗАДАНОЙ, УВЕЛИЧЬТЕ N И ПОВТОРИТЕ ВВОД'
390 PRINT 'КОЭФФИЦИЕНТЫ ПОЛИНОМА'
400 FOR I=1 TO N: PRINT!2.0!' A(' I-1')='!F1.9!Z(I): NEXT I
410 INPUT 'ВВЕДИТЕ X=' X: LET Q=0: FOR I=N TO 2 STEP -1
420 LET Q=(Q+Z(I))*X: NEXT I: LET Q=Q+Z(1)
430 PRINT 'ЗНАЧЕНИЕ Y(X)=' Q: GOTO 410: END

```

Пример. Используем пример, приведенный к программе 5.24. Так, если задать $E=1 \cdot 10^{-6}$, программа (после примерно 1 мин счета) выдает: $M=2$, $E=0$, $A(0)=3$, $A(1)=-0,4$, $A(2)=-0,01$ и $Y(X)=5,31$ при $X=7$. Время счета существенно больше, чем по программе 5.25, в связи с вычислением E и автоматическим выбором степени полинома m .

§ 5.7. Сглаживание данные эксперимента

Сглаживание данных эксперимента является специальной операцией усреднения с помощью интерполяционных полиномов, обеспечивающей получение уточненного значения \bar{y}_i по заданному значению y_i и ряду близлежащих значений $(\dots, y_{i-1}, y_i, y_{i+1}, \dots)$, известных со случайной погрешностью.

Линейное сглаживание по трем точкам реализуется с помощью следующих формул:

$$\begin{aligned} \bar{y}_0 &= (5y_0 + 2y_1 - y_2)/6, \\ \bar{y}_i &= (y_{i-1} + y_i + y_{i+1})/3, \quad 1 \leq i \leq N-1, \\ \bar{y}_N &= (5y_N + 2y_{N-1} - y_{N-2})/6, \end{aligned}$$

где N — номер последней точки (ординаты y_i).

Программа 5.27.

Пример сглаживания дан в табл. 5.2. *Линейное сглаживание по 5 точкам* проводится с использованием формул

$$\begin{aligned} \bar{y}_0 &= (3y_0 + 2y_1 + y_2 - y_4)/5, \\ \bar{y}_1 &= (4y_0 + 3y_1 + 2y_2 + y_3)/10, \\ \bar{y}_i &= (y_{i-2} + y_{i-1} + y_i + y_{i+1} + y_{i+2})/5, \\ & \quad 2 \leq i \leq N-2, \\ \bar{y}_{N-1} &= (y_{N-3} + 2y_{N-2} + 3y_{N-1} + 4y_N)/10, \\ \bar{y}_N &= (3y_N + 2y_{N-1} + y_{N-2} - y_{N-4})/5. \end{aligned}$$

```

10 PRINT!F1.9!'ЛИНЕЙНОЕ СГЛАЖИВАНИЕ ПО 3 ТОЧКАМ'
20 INPUT'ВВЕДИТЕ НОМЕР ПОСЛЕДНЕЙ ТОЧКИ N='N
30 PRINT'ВВЕДИТЕ Y 0':INPUT A
40 PRINT'ВВЕДИТЕ Y 1':INPUT B
50 PRINT'ВВЕДИТЕ Y 2':INPUT C
60 PRINT'SY 0='(5*A+2*B-C)/6
70 PRINT'SY 1='(A+B+C)/3
80 FOR I=2 TO N-1:LETA=B:LETB=C
90 PRINT!3.0!'ВВЕДИТЕ Y'I+1'=:INPUT C
100 PRINT'SY'I'='!1.9!(A+B+C)/3:NEXT I
110 PRINT!3.0!'SY'N'='!F1.9!(-A+2*B+5*C)/6
120 END
    
```

Таблица 5.2

Результаты линейного сглаживания по 3 и 5 точкам (N=9)

i	y_i	\bar{y}_i для сглаживания по 3 точкам	\bar{y}_i для сглаживания по 5 точкам	точное значение $y=10x$	i	y_i	y_i для сглаживания по 3 точкам	\bar{y}_i для сглаживания по 5 точкам	точное значение $y=10x$
0	0,9	0,97	0,992	1	5	6,1	5,973333333	6,044	6
1	2,12	1,98	1,995	2	6	6,92	7,056666666	7,024	7
2	2,92	3,063333333	2,998	3	7	8,15	8,04	8,004	8
3	4,15	3,99	4,038	4	8	9,05	9	8,957	9
4	4,9	5,05	4,998	5	9	9,8	9,825	9,91	10

Программа 5.28.

```

10 PRINT'ЛИНЕЙНОЕ СГЛАЖИВАНИЕ ПО 5 ТОЧКАМ'
20 INPUT'ВВЕДИТЕ НОМЕР ПОСЛЕДНЕЙ ТОЧКИ N='N
30 PRINT!F1.9!'ВВЕДИТЕ Y 0':INPUT A
40 PRINT'ВВЕДИТЕ Y 1':INPUT B
50 PRINT'ВВЕДИТЕ Y 2':INPUT C
60 PRINT'ВВЕДИТЕ Y 3':INPUT D
70 PRINT'ВВЕДИТЕ Y 4':INPUT E
80 PRINT'SY 0='(3*A+2*B+C-E)/5
90 PRINT'SY 1='(4*A+3*B+2*C+D)/10
100 PRINT'SY 2='(A+B+C+D+E)/5
110 FOR I=3 TO N-2
120 LETA=B:LETB=C:LETC=D:LETD=E
130 PRINT!3.0!'ВВЕДИТЕ Y'I+2'=:INPUT E
140 PRINT'SY'I'='!F1.9!(A+B+C+D+E)/5
150 NEXT I
160 PRINT!3.0!'SY'N-1'='!F1.9!(B+2*C+3*D+4*E)/10
170 PRINT!3.0!'SY'N'='!F1.9!(3*E+2*D+C-A)/5
180 END
    
```

Контрольный пример см. в табл. 5.2.

Нелинейное сглаживание по семи точкам

обеспечивает усреднение на основе применения полинома третьей степени и реализуется применением формул

$$\begin{aligned}
 y_0 &= (39y_0 + 8y_1 - 4(y_2 + y_3 - y_4) + y_5 - 2y_6) / 42, \\
 y_1 &= (8y_0 + 19y_1 + 16y_2 + 6y_3 - 4y_4 - 7y_5 + \\
 &\quad + 4y_7) / 42, \\
 y_2 &= (-4y_0 + 16y_1 + 19y_2 + 12y_3 + 2y_4 - 4y_5 + \\
 &\quad + y_6) / 42,
 \end{aligned}$$

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

$$\bar{y}_{N-2} = (y_{N-6} - 4y_{N-5} + 2y_{N-4} + 12y_{N-3} + 19y_{N-2} + 16y_{N-1} - 4y_N) / 42,$$

$$\bar{y}_{N-1} = (4y_{N-6} - 7y_{N-5} - 4y_{N-4} + 6y_{N-3} + 16y_{N-2} + 19y_{N-1} + 8y_N) / 42,$$

$$\bar{y}_N = (-2y_{N-6} + 4y_{N-5} + y_{N-4} - 4y_{N-3} - 4y_{N-2} + 8y_{N-1} + 39y_N) / 42.$$

```

10 PRINT'НЕЛИНЕЙНОЕ СГЛАЖИВАНИЕ ПО 7 ТОЧКАМ'
20 INPUT'ЗАДАЙТЕ НОМЕР ПОСЛЕДНЕГО ОТСЧЕТА N='N
30 INPUT'ВВЕДИТЕ ОТСЧЕТЫ Y0,Y1,...,Y6'A,B,C,D,E,F,G
40 PRINT'SY 0='(39*A+8*B-4*C-4*D+E+4*F-2*G)/42
50 PRINT'SY 1='(8*A+19*B+16*C+6*D-4*E-7*F+4*G)/42
60 PRINT'SY 2='(-4*A+16*B+19*C+12*D+2*E-4*F+G)/42
70 PRINT'SY 3='(-2*A+3*B+6*C+7*D+6*E+3*F-2*G)/21
80 FOR I=4 TO N-3:LETA=B:LETB=C:LETC=D:LETD=E
90 LETE=F:LETF=G:PRINT!3.0!'ВВЕДИТЕ Y'I+3'=':INPUT G
100 LETS=-2*A+3*B+6*C+7*D+6*E
110 PRINT!3.0!'SY'I'='!1.9!(S+3*F-2*G)/21:NEXT I
120 LETS=A-4*B+2*C+12*D+19*E+16*F-4*G
130 PRINT!3.0!'SY'N-2'='!F1.9! S/42
140 LETS=4*A-7*B-4*C+6*D+16*E+19*F+8*G
150 PRINT!3.0!'SY'N-1'='!F1.9! S/42
160 LETS=-2*A+4*B+C-4*D-4*E+8*F+39*G
170 PRINT!3.0!'SY'N'='!F1.9! S/42:END

```

Контрольный пример см. в табл. 5.3, грубо заданной нелинейной функции где представлены данные сглаживания $y_i = \ln(i+2)$.

Таблица 5.3

Результаты нелинейного сглаживания по семи точкам ($N=9$)

i	y_i	\bar{y}_i	Точное значение $y_i = \ln(i+2)$	i	y_i	\bar{y}_i	Точное значение $y_i = \ln(i+2)$
0	0,7	0,70114285714	0,6931471806	5	1,99	1,950952381	1,945910149
1	1,08	1,08714286	1,098612289	6	2,04	2,073333333	2,079441542
2	1,39	1,385	1,386294361	7	2,22	2,191428571	2,197224577
3	1,64	1,624285714	1,609437912	8	2,28	2,302857143	2,302585093
4	1,76	1,79952381	1,791759469	9	2,42	2,412619048	2,397895273

§ 6.1. Методы вычисления специальных функций

Специальными функциями обычно называются функции, которые в общем случае нельзя выразить через элементарные функции. Большинство из них являются решениями дифференциальных уравнений специального вида и могут быть представлены в виде интегралов. Однако эти интегралы нередко являются несобственными и имеют особенности подынтегральной функции на пределах или в промежутке между ними. В этой главе перечислены основные методы вычисления специальных функций.

Решение порождающих дифференциальных уравнений позволяет найти ряд значений специальных функций при заданных начальных условиях. Для этого могут использоваться программы, описанные в § 4.10. При описании специальных функций далее приводятся порождающие их дифференциальные уравнения. Однако этот метод (особенно при вычислениях с заданной точностью) сопряжен с применением сложных программ. Если требуется вычисление значений специальных функций при произвольных или часто сменяемых значениях аргумента, этот метод может привести к недопустимо большим затратам машинного времени.

Вычисление специальных функций численным интегрированием является универсальным и достаточно эффективным методом. Для его реализации могут использоваться программы численного интегрирования, описанные в § 4.9, 4.10. Следует учитывать, что в них аргумент подынтегральной функции обозначен через X , тогда как у специальных функций возможно иное обозначение, например t . Следовательно, в этом случае $t=X$. Подынтегральные выражения должны вычисляться по соответствующим подпрограммам (ввиду простоты тексты подпрограмм не приводятся). В целом программы вычисления специальных функций численным интегрированием достаточно громоздки и вычисления требуют значительных затрат машинного времени (в сравнении с изложенными ниже методами).

Вычисление специальных функций по рекуррентным соотношениям является средством, сравнительно быстрого получения численных значений нужных специальных функций. Примером может служить вычисле-

ние значений ортогональных многочленов в § 3.4. Однако этот метод применим при следующих условиях: имеются исходные значения функций или простые формулы для их вычисления, рекуррентный процесс является устойчивым (т. е. не ведет к накоплению ошибок), число циклов вычислений ограничено разумными пределами.

Вычисление специальных функций по их разложениям в ряд является в большинстве случаев наиболее удобным при машинных расчетах. Многие специальные функции имеют представления в виде сходящихся рядов. Вычисления должны вестись с контролем суммы членов ряда либо значений последнего члена (если есть гарантия, что они по модулю монотонно убывают) до тех пор, пока результат не будет получен с заданной погрешностью.

Вычисление специальных функций по их аппроксимациям является, как правило, методом, обеспечивающим получение результата за минимальное время. В большинстве случаев аппроксимация получается путем разложения функции в ряд (например, Тейлора) с удержанием конечного числа членов, достаточного для вычисления значений функции с заданной точностью. Сокращение времени вычислений обусловлено предварительным вычислением входящих в члены ряда выражений, например факториалов, значений тригонометрических функций и т. д. Однако достаточно простые аппроксимации получаются, как правило, при погрешности вычислений, заметно превосходящей предельную для ПЭВМ. Кроме того, вид аппроксимации обычно не имеет ничего общего с видом функции и для описания аппроксимации приходится использовать большое число численных констант с большим числом цифр. Поэтому вычисления по аппроксимациям следует рассматривать как дополнение к другим методам вычислений.

§ 6.2. Интегральные показательные функции

Интегральная показательная функция

$$E_n(x) = EN(X) = \int_1^{\infty} \frac{\exp(-xt)}{t^n} dt$$

вычисляется с помощью рекуррентного соотношения

$$E_{n+1}(x) = \frac{1}{n} [e^{-x} - xE_n(x)], \quad n=1, 2, 3, \dots$$

при $E_0(x) = \exp(-x)/x$ и с помощью функции $E_1(x)$, вычисляемой разложением в ряд:

$$E_1(x) = -\gamma - \ln x - \sum_{i=1}^{\infty} (-1)^i x^i / i \cdot i!$$

где $\gamma = 0,57721156647$ — постоянная Эйлера.

Программа 6.1.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ EN(X)'
```

```
10 INPUT 'ВВЕДИТЕ N=':N
```

```
20 INPUT 'ВВЕДИТЕ N=':N
```

```
30 INPUT 'ВВЕДИТЕ X=':X
```

```
40 LET K=EXP(-X):LETE=K/X
```

```
40 IF N>0 THEN 50
```

```
45 PRINT 'AN(X)=':A:GOTO 20
```

```
50 FOR I=1 TO N:LETA=(K+I*X)/X
```

```
60 NEXT I:PRINT 'AN(X)=':A
```

```
70 GOTO 20:END
```

Пример. Для $x=0,5$ имеем $E_0(0,5) = 1,213061319$, $E_1(0,5) = 0,5597736065$,
 $E_2(0,5) = 0,3266438565$, $E_{10}(0,5) = 0,0634583004$.

Интегральная показательная функция

$$Ei(x) = Ei(X) = -vp \int_0^{\infty} \frac{\exp(-t)}{t} dt, \quad x > 0,$$

вычисляется разложением в ряд

$$Ei(x) = \gamma + \ln x + \sum_{i=1}^{\infty} \frac{x^i}{i \cdot i!}.$$

Программа 6.2.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ EI(X)'
```

```
10 INPUT 'ВВЕДИТЕ N=':N
```

```
20 INPUT 'ВВЕДИТЕ X=':X
```

```
30 LET P=1:LETR=1:LETS=0:LETN=0
```

```
40 LETN=N+1:LETP=P*X:LETR=R*X
```

```
50 LETC=P/(N*X):LETS=S+C
```

```
60 IF C>N THEN 40
```

```
70 LETY=S+.5772156649+LOG(X)
```

```
80 PRINT 'EI(X)=':Y
```

```
90 GOTO 20:END
```

Пример. Для $x=0,5$ получим $Ei(0,5) = 0,4542199049$.

Функция

$$\alpha_n(x) = AN(X) = \int_1^{\infty} t^n e^{-xt} dt, \quad n=0, 1, 2, \dots,$$

вычисляется по рекуррентной формуле

$$\alpha_n(x) = [e^{-x} + n\alpha_{n-1}(x)]/x$$

при $\alpha_0(x) = e^{-x}/x$.

Программа 6.3.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ AN(X)'
```

```
10 INPUT 'ВВЕДИТЕ N=':N
```

```
20 INPUT 'ВВЕДИТЕ X=':X
```

```
30 LET K=EXP(-X):LETA=K/X
```

```
40 IF N>0 THEN 50
```

```
45 PRINT 'AN(X)=':A:GOTO 20
```

```
50 FOR I=1 TO N:LETA=(K+I*X)/X
```

```
60 NEXT I:PRINT 'AN(X)=':A
```

```
70 GOTO 20:END
```

Пример. Для $x=2$ получим $\alpha_0(2) = 0,0676676416$ и $\alpha_6(2) = 5,99497344$.

Функция

$$\beta_n(x) = BN(X) = \int_{-1}^1 t^n e^{-xt} dt, \quad n=0, 1, 2, \dots,$$

вычисляется по рекуррентной формуле

$$\beta_n(x) = [(-1)^n e^x - e^{-x} + n\beta_{n-1}(x)]/x$$

при $\beta_0(x) = (e^x - e^{-x})/x$.

Программа 6.4.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ BN(X)'
```

```
10 INPUT 'ВВЕДИТЕ N=':N
```

```
20 INPUT 'ВВЕДИТЕ X=':X
```

```
30 LET K=EXP(X):LETB=(K-1/K)/X
```

```
40 IF N>0 THEN 50
```

```
45 PRINT 'BN(X)=':B:GOTO 20
```

```
50 LETR=1:FOR I=1 TO N:LETR=-R
```

```
60 LETB=(R*(K-(1/K)+B*I))/X
```

```
70 NEXT I:PRINT 'BN(X)=':B
```

```
80 GOTO 20:END
```

Пример. Для $x=4$ получим $\beta_0(4) = 13,6449586$, $\beta_1(4) = -10,24287677$, $\beta_3(4) = -7,261476257$.

§ 6.3. Интегральные синус и косинус

Интегральный синус

$$Si(x) = SI(X) = \int_0^x \frac{\sin t}{t} dt$$

вычисляется разложением в ряд

$$Si(x) = \sum_{n=0}^{\infty} (-1)^n x^{2n+1} / (2n+1) (2n+1)!$$

Ряд прерывается, если абсолютная величина последнего члена становится меньше $\epsilon = 1 \cdot 10^{-9}$.

Программа 6.5.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ SI(X)'
```

```
10 INPUT 'ВВЕДИТЕ X=':X
```

```
20 LET B=X:LETC=X
```

```
30 LET D=(X^2)/2:LETI=0
```

```
40 LET I=I+1:LETE=(2*I+1)^2
```

```
50 LETB=((2*I-1)*D*B)/(I*E)
```

```
60 LETC=C+B: IF ABS(B)< 1E-9 THEN 80
```

```
70 GOTO 40
```

```
80 PRINT 'SI(N)=':C:GOTO 10:END
```

Пример. Для $x=0,1$ получим $Si(0,1) = 0,0999444611$, а для $x=10$ получим $Si(10) = 1,658347594$.

$$Ci(x) = CI(X) = \gamma + \ln x + \int_0^x \frac{\cos t - 1}{t} dt$$

$$\Gamma(x) \approx \sqrt{\frac{2\pi}{x}} e^{-x} x^x \left(1 + \frac{1}{12x}\right). \quad (6.1)$$

вычисляется разложением в ряд

$$Ci(x) = \gamma + \ln x + \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{2n \cdot (2n)!}$$

Программа 6.6.

```
05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИИ CI(X)'
10 INPUT 'ВВЕДИТЕ X=':X
20 LETS=X^2:LETI=0:LETA=0
30 LETB=1:LETC=1
40 LETI=I+1:LETD=2*I
50 LETC=(B-1)*D*(C): LETB=B*S
60 LETE=B/(C*D):LETA=A+E
70 IF ABS(E)<1E-9 THEN 90
80 GOTO 40
90 PRINT 'CI(X)=':A+.5772156649+LOG(X)
100 GOTO 10:END
```

Пример. Для $x=1$ получим $Ci(1) = 0,337403923$.

Вычисление $Si(x)$ и $Ci(x)$ численным интегрированием целесообразно лишь при малых значениях x . При больших $x > 8$ существенное уменьшение времени вычислений достигается при использовании асимптотических выражений [36].

§ 6.4. Гамма-функции (включая неполные)

Гамма-функция

$$\Gamma(x) = \Gamma(X) = \int_0^{\infty} e^{-t} t^{x-1} dt$$

является одной из наиболее распространенных специальных функций. Поэтому целесообразно рассмотреть различные алгоритмы ее вычислений.

Для $-18 < x < 49$ с погрешностью порядка $1 \cdot 10^{-6}$ $\Gamma(x)$ может быть вычислено с помощью 20-кратного преобразования по формуле $\Gamma(x+1) = x\Gamma(x)$ и с применением

Программа 6.7.

```
10 PRINT 'ВЫЧИСЛЕНИЕ ГАММА-ФУНКЦИИ Г(X)'
20 INPUT 'ВВЕДИТЕ X':Z:LETX=Z:LETA=1
30 FOR I=1 TO 20
40 LETZ=Z*(X+1):NEXT I
50 LETB=X+21
60 LETG=EXP(B*(LOG(B)-1)+1/12/B)*SQRT(2*PI/B)/Z
70 PRINT 'ЗНАЧЕНИЕ Г(X)=':G:GOTO 20:END
```

Программа 6.8.

```
10 PRINT 'ВЫЧИСЛЕНИЕ ГАММА-ФУНКЦИИ Г(X)'
20 INPUT 'ВВЕДИТЕ X':X:LETA=X-1:LETZ=A-64:IF Z<0 THEN 40
30 LETD=1:GOTO 60
40 LETB=ABS(INT(Z-1)):LETC=A:LETA=A+B:LETD=1:FOR I=B TO 1 STEP -1
50 LETC=C+1:LETD=D/C:NEXT I
60 LETG=((A/EXP(1))^A)*EXP(1/12/A)*SQRT(2*PI*A)*D
70 PRINT 'Г(X)=':G:GOTO 20:END
```

Такое преобразование необходимо для уменьшения погрешности (6.1). Алгоритм вычисления $\Gamma(x)$ при этом следующий.

1. Вводим x и полагаем $Z=x$, $X=Z$ и $A=1$.
2. Вычисляем $Z_l \leftarrow Z_{l-1}(X+1)$, где $l = 1, 2, \dots, 20$.
3. Полагаем $B=X+21$ и вычисляем $G = \exp(B(\ln B - 1) + 1/(12B)) \sqrt{2\pi/B} / Z_{20}$.

Программа 6.7.

Пример. Для различных x получаем $\Gamma(-1,5) = 2,363272685$ (2,363271801), $\Gamma(-0,5) = -3,544908843$ (-3,544907702), $\Gamma(1,5) = 0,8862271413$ (0,886226926), $\Gamma(40) = 2,039788231 \cdot 10^{46}$ (2,039788208 $\cdot 10^{46}$). В скобках указаны точные значения $\Gamma(x)$.

Более высокую точность при $-10 < x \leq 70$ имеет метод, в основе которого лежит следующий алгоритм.

1. Вводим $x=X$ и задаем $A=A-1$.
2. Полагаем $Z=A-64$.
3. Если $Z < 0$, идем к п. 4, иначе задаем $D=1$ и идем к п. 6.
4. Вычисляем $B = |\text{int}(Z-1)|$ и полагаем $C=A$, $A=A+B$ и $D=1$.
5. Находим

$$C_l = C_{l-1} + 1, \quad D_l = D_{l-1} / C,$$

где $l = B, B-1, B-2, \dots, 2, 1$.

6. Вычисляем $G = \Gamma(x)$ по формуле Стирлинга в следующей ее модификации:

$$G = \left(\frac{A}{e}\right)^A e^{(1/(12A))} D \sqrt{2\pi A}.$$

При этом алгоритме формула Стирлинга используется при аргументе, значения которого превышают 64, что дает меньшие погрешности. Однако при $x < -10$ корректирующий множитель $D < 1 \cdot 10^{-99}$ и ему, как и значению $\Gamma(x)$, присписывается нулевое значение.

Программа 6.8.

Пример. $\Gamma(-1,5) = 2,363271825$, $\Gamma(-0,5) = -3,544907737$, $\Gamma(1,5) = 0,8862269342$ и $\Gamma(40) = 2,039788229 \cdot 10^{46}$.

Наименшее время вычислений обеспечивается алгоритм, основанный на аппроксимации $\Gamma(x)$ на отрезке $[0, 1]$ степенным полиномом.

1. Из аргумента $B = |x|$ последовательно вычитаем 1 для того, чтобы B попал в отрезок $[0, 1]$. Одновременно подсчитываем произведение $D = B(B-1)(B-2) \dots (B-K)$, где K — число вычитаний 1 из B .

2. Если $x > 0$, вычисляем $\Gamma(x)$ по формулам
 $Y = B - K, F = \Gamma(Y + 1) =$
 $= ((((((0,035868343Y - 0,193527818)Y +$
 $+ (0,482199394)Y - 0,756704078)Y +$
 $+ (0,918206857)Y - 0,897056937)Y +$
 $+ (0,988205891)Y - 0,577191652)Y + 1,$
 $\Gamma(X) = DF/X.$

3. Если $x < 0$, вычисляем Y и F по п. 2 для $|x|$, после чего находим

$$\Gamma(x) = \frac{\pi}{\sin \pi x} \cdot \frac{1}{\Gamma(x-1)} = \frac{\pi}{\sin \pi x} \cdot \frac{1}{DF}.$$

Программа 6.9.

```
10 PRINT'ВЫЧИСЛЕНИЕ ГАММА-ФУНКЦИИ Г(Х) ПО АППРОКСИМАЦИИ'
20 INPUT'ВВЕДИТЕ X='X:LETB=ABS(X):LETD=1
30 IF B<=1 GOTO 50
40 LETD=D*B:LETB=B-1:GOTO 30
50 LETF=(((.035868343*B-.193527818)*B+.482199394)*B
60 LETF=((F-.756704078)*B+.918206857)*B-.897056937)*B
70 LETF=((F+.988205891)*B-.577191652)*B+1
80 LETG=F*D/X:IF X<0 GOTO 100
90 PRINT'Г(Х)='G:GOTO 20
100 LETG=#PI/SIN(#PI*X)/D/F:GOTO 90:END
```

Пример. $\Gamma(-3,2) = 0,6890562856$,
 $\Gamma(-2,5) = -0,9453086453$, $\Gamma(2) = 1$,
 $\Gamma(0,5) = 1,772453992$ и $\Gamma(50) = 6,082818641 \times 10^{62}$. Погрешность вычислений в этом методе не превышает $3 \cdot 10^{-7}$.

Логарифмическая производная гамма-функции (диагамма-функция)

$$\psi(x) = \frac{d}{dx} [\ln \Gamma(x)]$$

с погрешностью менее $2 \cdot 10^{-5}$ для $x > 20$ может вычисляться по асимптотическому разложению

$$\psi(x) = \ln x - 1/(2x) - 1/(12x^2) + \dots$$

С помощью рекуррентной формулы $\psi(x+1) = \psi(x) + 1/x$ значения $\psi(x)$ при $x < 20$ можно найти, вычисляя $\psi(x+20)$ и последовательно применяя эту формулу в направлении уменьшения x .

Программа 6.10.

```
10 PRINT'ВЫЧИСЛЕНИЕ ЛОГАРИФМИЧЕСКОЙ ПРОИЗВОДНОЙ'
15 PRINT'ГАММА-ФУНКЦИИ D(ЛОГ Г(Х))/DX'
20 INPUT'ВВЕДИТЕ X='X:IF X>20 THEN 40
30 LETC=20+(X-INT(X)):GOTO 50
40 LETX=X+1:LETC=X
50 LETA=1/2/C:LETA=-A*A/3-A+LOG(C)
60 LETC=C-1:LETA=A-1/C
70 IF X<C THEN 60
80 PRINT'ЗНАЧЕНИЕ D(ЛОГ Г(Х))/DX='A
90 GOTO 20:END
```

Программа 6.11.

```
10 PRINT'ВЫЧИСЛЕНИЕ НЕПОЛНОЙ ГАММА-ФУНКЦИИ G(A,X)'
20 INPUT'ВВЕДИТЕ A,X 'A,X:LETB=INT(2*X+9):LETY=B
30 FOR I=B TO 1 STEP -1:LETY=Y*(I+A)+1:NEXT I
40 LETG=Y*EXP(-X)*X^A/A:PRINT'G(A,X)='G:GOTO 20:END
```

Пример. $\psi(1,5) = 0,03648992674$,
 $\psi(50) = 3,901989672$.
 Неполные гамма-функции задаются соотношениями

$$P(a, x) = \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt,$$

$$\gamma(a, x) = P(a, x) \Gamma(a) = \int_0^x e^{-t} t^{a-1} dt, \quad (6.2)$$

$$\Gamma(a, x) = \Gamma(a) - \gamma(a, x) = \int_0^\infty e^{-t} t^{a-1} dt,$$

$$\gamma^*(a, x) = x^a P(a, x) = x^{-a} \gamma(a, x) / \Gamma(a).$$

В качестве основной можно выбрать функцию $\gamma(a, x)$, легко вычисляемую разложе-

нием в ряд ($x \geq 0, a > 0$)

$$\gamma(a, x) = e^{-x} \frac{x^a}{a} \left[\sum_{i=0}^N \frac{x^i}{(a+1)(a+2)\dots(a+i)} + \theta_{N+1} \right],$$

где $N = \text{int}(2x+8)$ и θ_{N+1} — корректирующий член. Зная $\gamma(a, x)$ и $\Gamma(a)$, можно вычислить любую другую из приведенных неполных гамма-функций. Вычисление $\gamma(a, x)$ реализуется программой 6.11.

Программа 6.11.

Пример. $\gamma(2; 8) = 0,9969825948$,
 $\gamma(0,5; 1) = 1,493648313$.

Неполная гамма-функция $\gamma(a, x)$ может вычисляться также численным интегрирова-

нием — см. формулу (6.2). Так, используя программу 4.42 численного интегрирования методом Гаусса и задав $a=2$, $B=x=8$, $A=0$ и $M=8$, получим $\gamma(a, x) = 0,9969832028$ при $t_c \approx 20$ с.

§ 6.5. Функции Бесселя (включая модифицированные)

Функции Бесселя являются решениями линейного дифференциального уравнения второго порядка

$$\frac{d^2\omega}{dx^2} + \frac{1}{x} \frac{d\omega}{dx} + \left(1 - \frac{\nu^2}{x^2}\right)\omega = 0.$$

Здесь ν — порядок функции. Функции Бесселя могут быть первого $J_{\pm\nu}(x)$, второго $Y_{\pm\nu}(x)$ и третьего $H_{\pm\nu}(x)$ родов. Они связаны следующими соотношениями:

$$Y_\nu(x) = (J_\nu(x) \cos(\nu\pi) - J_{-\nu}(x)) / \sin(\nu\pi), \quad (6.3)$$

$$H_{\nu,2}^\nu(x) = J_\nu(x) \pm j Y_\nu(x).$$

Соотношение (6.3) справедливо при дробном порядке ν . При целых ν оно заменяется предельным переходом.

Решениями дифференциального уравнения

$$\frac{d^2\omega}{dx^2} + \frac{1}{x} \frac{d\omega}{dx} - \left(1 + \frac{\nu^2}{x^2}\right)\omega = 0$$

являются модифицированные функции Бесселя $I_\nu(x)$ и $K_\nu(x)$:

$$K_\nu(x) = \frac{\pi}{2} \frac{I_{-\nu}(x) - I_\nu(x)}{\sin(\nu\pi)}. \quad (6.4)$$

Функции Бесселя первого рода $J_n(x)$ и $I_n(x)$ при целом порядке $\nu=n$ ($n=0, 1, 2, \dots$) могут вычисляться разложениями в ряды

$$J_n(x) = \left(\frac{x}{2}\right)^n \sum_{i=0}^{\infty} \frac{(-x^2/4)^i}{i!(i+n)!},$$

$$I_n(x) = \left(\frac{x}{2}\right)^n \sum_{i=0}^{\infty} \frac{(x^2/4)^i}{i!(i+n)!}.$$

Ряды для $J_n(x)$ и $I_n(x)$ различаются лишь знаком у члена $(x^2/4)^i$. Это позволяет вычислять данные функции по одной программе.

Программа 6.12.

```

05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ JN(X) И IN(X)'
10 INPUT 'ВВЕДИТЕ -1 ДЛЯ JN(X) ИЛИ 1 ДЛЯ IN(X)':'Z
20 INPUT 'ВВЕДИТЕ N=':N
30 INPUT 'ВВЕДИТЕ X=':X
40 LETS=0:LETA=1
50 IF N<=1 THEN 70
60 FOR I=1 TO N:LETA=A*I:NEXT I
70 LETY=(X/2)^2:LETI=0:LETS=1:LETC=1:LETR=1
80 LETI=I+1:LETS=I*B:LETC=(N+I)*C
90 LETR=Z*R:LETE=(Y^I)*R/B/C:LETS=S+E
100 IF ABS(E)>1E-9 THEN 80
110 LETJ=(1+S)*((X/2)^N)/A
120 IF Z<0 THEN 140
130 PRINT 'JN(X)=':J:GOTO 30
140 PRINT 'IN(X)=':J:GOTO 30:END

```

Пример. $J_0(0,5) = 0,9384698072$, $J_{30}(20) = 1,240153633 \cdot 10^{-4}$, $I_0(2) = 2,279585302$ и $I_1(2) = 1,590636855$.

Функции Бесселя $J_\nu(x)$ и $I_\nu(x)$ при дробном порядке ν вычисляются по следующим разложениям в ряд:

$$J_\nu(x) = \left(\frac{x}{2}\right)^\nu \sum_{K=0}^{\infty} \frac{(-x^2/4)^K}{K! \Gamma(\nu+K+1)},$$

$$I_\nu(x) = \left(\frac{x}{2}\right)^\nu \sum_{K=0}^{\infty} \frac{(x^2/4)^K}{K! \Gamma(\nu+K+1)}.$$

Для вычисления функции $\Gamma(\nu+K+1) = \Gamma(\nu) \cdot \nu(\nu+1)(\nu+2) \dots (\nu+K)$ должна использоваться часть программы для вычисления гамма-функции, например, по аппроксимации. Вычислив функции $I_{\pm\nu}(x)$ и $J_{\pm\nu}(x)$ по формулам (6.3) и (6.4), находим функции Бесселя второго рода $Y_\nu(x)$ и $K_\nu(x)$.

Программа 6.13.

Пример. Для $\nu=V=1/3$ и $x=1,5$, задав код 0, получим $\Gamma(\nu) = 2,678938244$, $J_\nu(x) = 0,6371327061$, $\Gamma(-\nu) = -4,062354258$, $J_{-\nu}(x) = 0,2348995028$ и $Y_\nu(x) = 0,09661015701$. Задав код 1, получим $I_\nu(x) = 1,501429163$, $I_{-\nu}(x) = 1,62280812$ и $K_\nu(x) = 0,2201570763$.

Вычисление функций Бесселя $Y_\nu(x)$ и $K_\nu(x)$ при целых $\nu=n$ невозможно. На практике приближенные значения этих функций можно найти, задав $\nu=n \pm \delta$, где δ — малая величина ($\delta \approx 1 \cdot 10^{-3} \div 1 \cdot 10^{-4}$). Результат можно уточнить, взяв среднее при $\nu = n + \delta$ и $\nu = n - \delta$. Следует отметить, что при $\nu \geq 0$ значения $x \leq 9$ для функций $I_\nu(x)$, при этом погрешность вычислений не превышает $2 \cdot 10^{-5}$ (с ростом ν диапазон значений x расширяется и при $\nu = 100$ $x \leq 50$). Погрешность вычисления $Y_\nu(x)$ и $K_\nu(x)$ выше; например, в приведенном примере $Y_\nu(x)$ верны 4 цифры после запятой.

Частными решениями дифференциального уравнения

$$\frac{d^2\omega}{dt^2} + \frac{2}{x} \frac{d\omega}{dt} + \left[1 - \frac{n(n+1)}{x^2}\right]\omega = 0,$$

$$n=0, \pm 1, \pm 2, \dots,$$

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ БЕССЕЛЯ '
15 PRINT '  J0(X), Y0(X), I0(X), K0(X) '
20 INPUT '0 ДЛЯ J0(X), Y0(X) И 1 ДЛЯ I0(X), K0(X) ? : K
30 INPUT 'ВВЕДИТЕ ПОРЯДОК ФУНКЦИИ U=' : U
40 INPUT 'ВВЕДИТЕ АРГУМЕНТ ФУНКЦИИ X=' : X
50 LET U=X*U/4: IF K=0 THEN LET U=-U
60 GOSUB 200: LET J=F
70 PRINT 'ЗНАЧЕНИЕ ГАММА-ФУНКЦИИ Γ(U)=' : G
80 IF K=1 THEN 100
90 PRINT 'J0(X)=' : J: GOTO 110
100 PRINT 'I0(X)=' : J
110 LET U=-U: GOSUB 200
120 PRINT 'ЗНАЧЕНИЕ ГАММА-ФУНКЦИИ Γ(-X)=' : G
130 LET A=-U*PI: IF K=1 THEN 170
140 PRINT 'J-U(X)=' : F
150 LET Y=J*COS(A)/SIN(A)-F/SIN(A)
160 PRINT 'Y0(X)=' : Y: GOTO 40
170 PRINT 'I-U(X)=' : F
180 LET K=PI*(F-J)/2/SIN(A)
190 PRINT 'K0(X)=' : K: GOTO 40
200 LET B=ABS(U): LET I=1
210 IF B<1 THEN 230
220 LET I=D*B: LET B=B-1: GOTO 210
230 LET F=((0.035868343*B-.193527818)*B+.482199394)*B
240 LET F=((F-.756704078)*B+.918206857)*B-.897056937)*B
250 LET F=((F+.988205891)*B-.577191652)*B+1
260 LET G=F*D/U: IF U=0 THEN 280
270 LET G=PI/SIN(PI*U)/D/F
280 LET F=1/G/U: LET R=F: LET I=0
290 LET I=I+1: LET R=R*U/I/(U+I): LET U=F
300 LET F=F+R: IF ABS(F-U)<>0 THEN 290
310 LET F=F*(X/2)^U: RETURN: END

```

являются сферические функции Бесселя первого рода

$$j_n(x) = \sqrt{\pi/(2x)} J_{n+1/2}(x),$$

второго рода

$$y_n(x) = \sqrt{2\pi/(2x)} Y_{n+1/2}(x)$$

и третьего рода

$$h_n(x) = \sqrt{2\pi/(2x)} H_{n+1/2}(x).$$

Эти функции можно найти по программам, вычисляющим функции $J_\nu(x)$ и $Y_\nu(x)$ при $\nu = n + 1/2$. Вычисления облегчаются тем, что при этом достаточно располагать известным значением $\Gamma(1/2) = \sqrt{\pi}$.

Пример (в скобках указаны точные значения). $j_0(2) = 0,4546487134$ (0,45464871), $j_8(10) = 0,1255780237$ (0,12558), $j_{10}(1) = -7,11655264 \cdot 10^{-11}$ (верны все цифры), $y_0(1) = -0,5403023059$, $y_2(2) = -0,7339914247$, $y_{10}(5) = -26,6561144$. Время счета от 3 до 12 с.

Функции Бесселя могут вычисляться также по их аппроксимациям степенными полиномами (см. [36]). Однако программы вычисления при этом получаются громоздкими из-за необходимости применения одновременно нескольких полиномов при изменении аргумента x от 0 до ∞ .

Программа 6.14.

```

10 PRINT 'ВЫЧИСЛЕНИЕ СФЕРИЧЕСКИХ ФУНКЦИЙ '
20 PRINT '  БЕССЕЛЯ JN(X) И YN(X) '
30 INPUT 'ВВЕДИТЕ X=' : X: LET A=1
40 INPUT 'ВВЕДИТЕ N=' : N
50 FOR N=0 TO N: LET A=A/(2*N+1): NEXT N
60 LET S=1: LET C=0: LET D=-X*X/2: LET F=1
70 LET C=C+1: LET B=B*C: LET E=2*C+1
80 LET F=F*D/C/(2*U+E): LET U=S
90 LET S=S+F: IF ABS(S-U)<>0 THEN 70
100 LET J=A*S*X^N: PRINT 'JN(X)=' : J
110 LET A=1: FOR N=0 TO N
120 LET A=A*(2*N-1): NEXT N
130 LET S=1: LET C=0: LET F=1
140 LET C=C+1: LET E=2*C-1: LET F=F*D/C/(E-2*U)
150 LET U=S: LET S=S+F: IF ABS(U-S)<>0 THEN 140
160 LET Y=A*S/(X^(U+1)): PRINT 'YN(X)=' : Y
170 GOTO 20: END

```

§ 6.6. Функции Эйри

Решению дифференциального уравнения

$$\frac{d^2\omega}{dt^2} - x\omega = 0$$

удовлетворяет пара независимых решений $A_i(x)$ и $B_i(x)$, являющихся функциями Эйри.

Функции Эйри могут вычисляться разложением в степенные ряды

$$A_i(x) = c_1 f(x) - c_2 g(x),$$

$$B_i(x) = \sqrt{3} [c_1 f(x) + c_2 g(x)],$$

$$f(x) = 1 + \frac{1}{3!}x^3 + \frac{1 \cdot 4}{6!}x^6 + \frac{1 \cdot 4 \cdot 7}{9!}x^9 + \dots$$

$$g(x) = x + \frac{2}{4!}x^4 + \frac{2 \cdot 5}{7!}x^7 + \frac{2 \cdot 5 \cdot 8}{10!}x^{10} + \dots$$

где $c_1 = 0,355028054$, $c_2 = 0,258819404$. Поскольку для $x < 0$ члены рядов оказываются знакопеременными, при $x < -7$ сходимость их резко ухудшается.

Программа 6.15.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ ЭЙРИ AI(X) И BI(X)'
20 INPUT 'ВВЕДИТЕ ПОГРЕШНОСТЬ E='E
30 INPUT 'ВВЕДИТЕ X='X:LET I=0
40 LET F=1:LETK=1:LETL=X^3:LETJ=1:LETR=1
50 LET I=I+3:LETK=K*L*J:LETM=F:LETR=R*I*(I-1)*(I-2)
60 LET F=F+K/R:LETJ=J+3:IF ABS(F-M)>E THEN 50
70 LET I=1:LETG=1:LETK=1:LETR=1
80 LET I=I+3:LETK=K*(I-2)*L:LETM=G:LETR=R*I*(I-1)*(I-2)
90 LET G=G+K/R:IF ABS(G-M)>E THEN 80
100 LET E=C*F-D*G:LETB=(C*F+D*G)*SQR(3)
110 LET A=C*F-D*G:LETB=(C*F+D*G)*SQR(3)
120 PRINT 'AI(X)='A:PRINT 'BI(X)='B:GOTO 30:END
    
```

Пример. При погрешности $E = 1 \cdot 10^{-6}$ получим $A_i(1) = 0,1352924154$, $B_i(1) = 1,207423594$ ($t_c \approx 3$ с), $A_i(-5) = 0,3507610391$ и $B_i(-5) = -0,1383692054$ ($t_c \approx 18$ с).

Для вычисления $C(x)$ можно применить программу 4.42, вписав в нее подпрограмму вычисления подынтегральной функции $\cos(\pi t^2/2)$ при $t=X$. Задав $A=0$, $B=x=2$ и $M=8$, получим $C(2) = 0,4882522862$ при точном значении $0,4882534$.

Для вычисления $S(x)$ также можно использовать программу 4.42. Задав $A=0$, $B=x=2$ и $M=8$, получим $S(2) = 0,3434145829$ при точном значении $0,3434157$.

При $x < 3$ для вычисления $C(x)$ и $S(x)$ могут использоваться следующие разложения в ряд:

$$C(x) = \sum_{n=0}^{\infty} \frac{(-1)^n (\pi/2)^{2n}}{(2n)!(4n+1)} x^{4n+1}$$

$$S(x) = \sum_{n=0}^{\infty} \frac{(-1)^n (\pi/2)^{2n+1}}{(2n+1)!(4n+3)} x^{4n+3}$$

При $x > 3$ эти ряды плохо сходятся и для вычисления $C(x)$ и $S(x)$ целесообразно

использовать асимптотические приближения

$$C(x) = \frac{1}{2} + \frac{\sin(\pi x^2/2)}{\pi x} \left(1 - \frac{3}{(\pi x^2)^2}\right) - \frac{\cos(\pi x^2/2)}{\pi^2 x^3} \left(1 - \frac{5}{(\pi x^2)^2}\right),$$

$$S(x) = \frac{1}{2} - \frac{\cos(\pi x^2/2)}{\pi x} \left(1 - \frac{3}{(\pi x^2)^2}\right) - \frac{\sin(\pi x^2/2)}{\pi^2 x^3} \left(1 - \frac{5}{(\pi x^2)^2}\right).$$

Погрешность вычислений не превышает $5/(\pi^4 x^7)$, что при $x \geq 4$ дает не менее 5 верных знаков после запятой.

Программа 6.16.

```

05 PRINT 'ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ ФРЕНЕЛЯ C(X) И S(X)'
10 INPUT 'ВВЕДИТЕ X='X:LETA=#PI*X^2
20 IF X>3 THEN 150
30 LETS=0:LETC=0:LETH=0:LETX=-1:LETR=1
40 GOTO 80
50 IF N>1 THEN 70
60 LETR=2:GOTO 80
70 LETR=R*2*N*(2*N-1)
    
```

§ 6.7. Интегралы Френеля

Интегралами Френеля называются интегральные функции

$$C(x) = \int_0^x \cos\left(\frac{\pi}{2} t^2\right) dt, \quad S(x) = \int_0^x \sin\left(\frac{\pi}{2} t^2\right) dt.$$

Они могут вычисляться непосредственно численным интегрированием.

```

80 LETZ=-Z: LETY=4*N
90 LETB=Z*((#PI/2)^(2*N))*C*(Y)/R
100 LETD=B/(4*N+1): LETC=C+D
110 LETE=B/(4*N+3)/(2*N+1): LETS=S+E
120 LETN=N+1: IF ABS(D)>1E-10 THEN 50
130 LETC=C*X: LETS=S*#PI*(X^3)/2
140 PRINT'C(X)='C:PRINT'S(X)='S:GOTO 10
150 LETE=(SIN(A/2))/#PI/X: LETF=(COS(A/2))/#PI/X
160 LETK=1-3/(A^2): LETL=1-5/(A^2)
170 LETC=.5+E*K-F*L/A
180 LETS=.5-F*K-E*L/A:GOTO 140:END

```

Пример. $C(0,2) = 0,1999210576$, $S(0,2) = 0,004187609162$ ($t_c \approx 6с$), $C(2) = 0,4882534059$, $S(2) = 0,3434156788$ (время счета $\approx 20 с$), $C(4) = 0,4984199894$, $S(4) = 0,4205170154$ (время счета менее 2 с).

§ 6.8. Эллиптические интегралы

Полные эллиптические интегралы первого $K(m)$ и второго $E(m)$ рода определяются как

$$K(m) = \int_0^{\pi/2} (1 - m \sin^2 \theta)^{-1/2} d\theta,$$

$$E(m) = \int_0^{\pi/2} (1 - m \sin^2 \theta)^{1/2} d\theta$$

и могут вычисляться непосредственно с помощью численного интегрирования.

Для вычисления $K(m)$ можно использовать программу 4.42. Задав $m=1/2$, $A=0$, $B=\pi/2$ и $M=4$, получим $K(1/2) = 1,854074665$ при точном значении $1,8540746733$ и $t_c \approx 15 с$.

Вычисление $E(m)$ также выполняется численным интегрированием. Задав $m=1/2$, $A=0$, $B=\pi/2$ и $M=4$, получим $E(1/2) = 1,350643882$ при точном значении $1,350643881$.

Эллиптические интегралы $K(m)$ и $E(m)$ могут вычисляться также по их разложениям в ряд:

$$K(m) = \frac{\pi}{2} \left[1 + \left(\frac{1}{2}\right)^2 m + \left(\frac{1 \cdot 3}{2 \cdot 4}\right)^2 m^2 + \dots \right].$$

$$E(m) = \frac{\pi}{2} \left[1 - \left(\frac{1}{2}\right)^2 \frac{m}{1} - \left(\frac{1 \cdot 3}{2 \cdot 4}\right)^2 \frac{m^2}{3} - \left(\frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}\right)^2 \frac{m^3}{5} - \dots \right].$$

Программа 6.17.

```

01 PRINT'ВЫЧИСЛЕНИЕ ПОЛНЫХ ЭЛЛИПТИЧЕСКИХ ИНТЕГРАЛОВ K(M) И E(M)'
05 INPUT'ИНТЕГРАЛ ЭЛЛИПТ. КАКОГО РОДА 1 ИЛИ 2 ?'C
10 INPUT'ВВЕДИТЕ M='M: LETI=0
20 LETA=1: LETB=1: LETS=1: LETR=1
30 LETI=I+1: LETA=A*(2*I-1): LETB=B*I*2
35 IF C=1 THEN 45
40 LETC=-1/(2*I-1)
45 LETR=M*R: LETD=C*R*(A/B)^2
50 LETS=S+D: IF ABS(D)>1E-9 THEN 30
60 LETK=#PI*S/2: IF C=1 THEN 80
70 PRINT'E(M)='K: GOTO 10
80 PRINT'K(M)='K: GOTO 10: END

```

Пример. $K(0,1) = 1,612441349$ ($t_c \approx 6 с$), $K(0,5) = 1,854074676$ ($t_c \approx 20 с$), $E(0,2) = 1,4890335058$ ($t_c \approx 9 с$).

С погрешностью менее $2 \cdot 10^{-8}$ полные эллиптические интегралы $K(m)$ вычисляются с помощью полиномиальной аппроксимации [36]:

$$K(m) = (((a_4 m_1 + a_3) m_1 + a_2) m_1 + a_1) m_1 + a_0 + (((b_4 m_1 + b_3) m_1 + b_2) m_1 + b_1) m_1 + b_0 \ln(1/m_1),$$

где $a_1 = 0,44325141463$, $a_2 = 0,0626060122$, $a_3 = 0,09666344259$, $a_4 = 0,03590092383$, $a_5 = 0,03742563713$, $a_6 = 0,01451196212$, $b_0 = 0,5$, $b_1 = 0,12498593597$, $b_2 = 0,06880248576$, $b_3 = 0,03328355346$, $b_4 = 0,00441787012$.

Аналогично

$$E(m) = (((a_4 m_1 + a_3) m_1 + a_2) m_1 + a_1) m_1 + 1 + (((b_4 m_1 + b_3) m_1 + b_2) m_1 + b_1) m_1 \ln(1/m_1),$$

где $a_1 = 0,44325141463$, $a_2 = 0,0626060122$, $a_3 = 0,04757383546$, $a_4 = 0,01736506451$, $b_1 = 0,24998368310$, $b_2 = 0,09200180037$, $b_3 = 0,040669697526$ и $b_4 = 0,00526449639$.

Программа 6.18.

Пример (в скобках даны точные значения). $K(0,1) = 1,6124441352$ ($1,612441349$), $E(0,1) = 1,530757641$ ($1,530757637$), $K(0,9) = 2,578092105$ ($2,578092113$), $E(0,9) = 1,104774722$ ($1,104774733$).

Неполные эллиптические интегралы

$$F(\varphi|m) = \int_0^{\varphi} (1 - m \sin^2 \theta)^{-1/2} d\theta,$$

$$E(\varphi|m) = \int_0^{\varphi} (1 - m \sin^2 \theta)^{1/2} d\theta$$

также могут вычисляться по программам численного интегрирования. В этом случае полагают $B=\varphi$.

Программа 6.18.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ПОЛНЫХ ЭЛЛИПТИЧЕСКИХ ИНТЕГРАЛОВ K(M) И E(M)'
15 PRINT '      С ПРИМЕНЕНИЕМ АППРОКСИМАЦИИ'
20 INPUT 'ВВЕДИТЕ M<=1 M='M:LETM=1-M
30 LETK=((0.1451196212*M+.03742563713)*M+.03590092383)*M
35 LETK=(K+.09666344259)*M+1.38629436112
40 LETW=(.00441787012*M+.03328355346)*M
45 LETW=(W+.06880248576)*M+.12498593597)*M+.5
50 LETK=K+W*LOG(1/M):PRINT 'K(M)='K
60 LETE=(.01736506451*M+.04757383546)*M
65 LETE=(E+.0626060122)*M+.44325141463)*M+1
70 LETW=(.00526449639*M+.04069697526)*M
75 LETW=(W+.09200180037)*M+.2499836831)*M
80 LETE=E+W*LOG(1/M):PRINT 'E(M)='E:GOTO 20:END
    
```

§ 6.9. Функции Струве, Ангера и Вебера

Функции Струве $H_\nu(x)$ входят в общее решение $\omega = aJ_\nu(x) + bY_\nu(x) + H_\nu(x)$ дифференциального уравнения второго порядка вида

$$x^2 \frac{d^2 \omega}{dx^2} + x \frac{d\omega}{dx} + (x^2 - \nu^2) \omega = \frac{4(x/2)^{\nu+1}}{\sqrt{\pi} \Gamma(\nu+1/2)}$$

Функции $H_\nu(x)$ вычисляются разложением в ряд [42]:

$$H_\nu(x) = \frac{2}{\sqrt{\pi}} \frac{(x/2)^{\nu+1}}{\Gamma(\nu+3/2)} \left[1 - \frac{x^2}{3 \cdot (2\nu+3)} + \frac{x^4}{3 \cdot 5 \cdot (2\nu+3) \cdot (2\nu+5)} - \dots \right]$$

Модифицированные функции Струве $L_\nu(x) = -je^{-j\pi\nu/2} H_\nu(jx)$ можно разложить в ряд вида

$$L_\nu(x) = \frac{2}{\sqrt{\pi}} \frac{(x/2)^{\nu+1}}{\Gamma(\nu+3/2)} \left[1 + \frac{x^2}{3 \cdot (2\nu+3)} + \frac{x^4}{3 \cdot 5 \cdot (2\nu+3) \cdot (2\nu+5)} + \dots \right]$$

Таким образом, разложения в ряд $H_\nu(x)$ и $L_\nu(x)$ различаются лишь тем, что члены в квадратных скобках у ряда $H_\nu(x)$ знакопередающиеся, а у ряда $L_\nu(x)$ — с одним

знаком. В программу для вычисления этих функций нужно включать фрагмент, вычисляющий $\Gamma(\nu+1,5)$, например, по аппроксимации гамма-функции полиномом.

Программа 6.19.

Пример. Погрешность $E = 1 \cdot 10^{-7}$, $H_1(5) = 0,8078118816$, $H_0(0,5) = 0,30955589$, $H_{-8}(4) = -8,249323233$, $L_1(0,5) = 0,05394217833$ и $L_{-6}(6) = 3,327822061$.

Функции Струве могут вычисляться также с помощью численного интегрирования с применением интегральных представлений ($\text{Re } \nu > -1/2$):

$$H_\nu(x) = \frac{2(x/2)^\nu}{\sqrt{\pi} \Gamma(\nu+1/2)} \int_0^1 (1-t^2)^{\nu-1/2} \sin(xt) dt,$$

$$L_\nu(x) = \frac{2(x/2)^\nu}{\sqrt{\pi} \Gamma(\nu+1/2)} \int_0^{\pi/2} \text{sh}(x \cos \theta) \sin^{2\nu} \theta d\theta.$$

Однако и в этом случае необходимо вычисление гамма-функции $\Gamma(\nu+1/2)$.

Функции Ангера $J_\nu(x)$ имеют интегральное представление вида

$$J_\nu(x) = \frac{1}{\pi} \int_0^\pi \cos(\nu\theta - x \sin \theta) d\theta.$$

При целом $\nu = n$ $J_\nu(x) = J_n(x)$. В общем

Программа 6.19.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ СТРУВЕ HU(X) И LV(X)'
15 INPUT 'ВВЕДИТЕ 0 ДЛЯ HU(X) И 1 ДЛЯ LV(X) 'K
20 INPUT 'ВВЕДИТЕ U, X = 'U, Y: LETA=Y*X: IF K=0 THEN LETA=-A
30 LETX=U+1.5: LETH=1: LETI=1: LETJ=1
40 LETB=ABS(X): LETD=1
50 IF B<1 THEN 70
60 LETD=D*B: LETB=B-1: GOTO 50
70 LETF=((0.35868343*B-.193527818)*B+.482199394)*B
80 LETF=((F-.756704078)*B+.918206857)*B-.897056937)*B
90 LETF=((F+.988205891)*B-.577191652)*B+1
100 LETG=F*D/X: IF X<0 THEN LETG=#PI/SIN(#PI*X)/D/F
110 LETI=I+2: LETH=H*A/I/(2*U+1): LETW=J
120 LETJ=J+H: IF ABS(J-W)>=1E-10 THEN 110
130 LETW=((Y/2)^(U+1))*2*J/6/SQR(#PI)
140 IF K=1 THEN 160
150 PRINT 'HU(X)='W: GOTO 20
160 PRINT 'LV(X)='W: GOTO 20:END
    
```


случае функций Ангера удобно вычислять с помощью численного интегрирования. Так, для $v=1, A=0, B=\pi, x=0,5, M=4$ получим $J_1(0,5) = 0,5844468394$ при $t_c \approx 10$ с.

Аналогичным образом несложно организовать вычисления функций Вебера

$$E_\nu(x) = \frac{1}{\pi} \int_0^\pi \sin(\nu\theta - x \sin \theta) d\theta.$$

§ 6.10. Гипергеометрические функции

Гипергеометрические функции $F(a, b, c, x)$ являются частным решением дифференциального уравнения

$$x(1-x) \frac{d^2 \omega}{dx^2} + [c - (a+b+1)x] \frac{d\omega}{dx} - ab\omega = 0.$$

Они могут вычисляться с применением следующего ряда:

$$F(a, b, c, x) \approx 1 + \sum_{i=0}^N \left[\prod_{j=0}^{i-1} \frac{(a+j)(b+j)}{(1+j)(c+j)} x \right] + \theta_{N+1},$$

где $N=22, \theta_{N+1}$ — поправка. Значения $|x| < 1, c \neq 0, -1, -2, \dots$. При $0 < (a, b) \leq c$ и $-0,1 \leq x \leq 0,8$ погрешность $\delta < 5 \cdot 10^{-2}$, но обычно существенно ниже.

Программа 6.20.

```
10 PRINT 'ВЫЧИСЛЕНИЕ ГИПЕРГЕОМЕТРИЧЕСКОЙ ФУНКЦИИ F(A,B,C,X)'
20 INPUT 'ВВЕДИТЕ A,B,C':A,B,C
30 INPUT 'ВВЕДИТЕ X=':X:LET F=LGT(23)
40 FOR I=23 TO 1 STEP -1:LETD=I-1
50 LET F=F*(D+A)*(D+B)/(D+C)/I*X+1
60 NEXT I:PRINT 'F(A,B,C,X)=':F:GOTO 30:END
```

Пример. $F(1; 1; 2; 0,5) = 1,386294358$ при точном значении $F(1; 1; 2; x) = -x^{-1} \ln(1-x) = 1,386294361$ и $t_c \approx 8$ с.

Первым независимым решением уравнения Куммера

$$x \frac{d^2 \omega}{dx^2} + (b-x) \frac{d\omega}{dx} - a\omega = 0$$

является вырожденная гипергеометрическая функция $F(a, b, x) \equiv M(a, b, x)$. Через нее и гамма-функцию может быть найдено второе

Программа 6.21.

```
10 PRINT 'ВЫЧИСЛЕНИЕ ВЫРОЖДЕННОЙ ГИПЕРГЕОМЕТРИЧЕСКОЙ'
15 PRINT '    ФУНКЦИИ F(A,B,X)'
20 INPUT 'ВВЕДИТЕ A,B':A,B
30 INPUT 'ВВЕДИТЕ X=':X:LET F=LGT(23)
40 FOR I=23 TO 1 STEP -1:LETD=I-1
50 LET F=F*(D+A)*X/(D+B)/I+1
60 NEXT I:PRINT 'ЗНАЧЕНИЕ F(A,B,X)=':F
70 GOTO 30:END
```

$$U(a, b, x) = \frac{\pi}{\sin \pi b} \left\{ \frac{F(a, b, x)}{\Gamma(1+a+b) \cdot \Gamma(b)} - x^{1-b} \frac{F(1+a-b, 2-b, x)}{\Gamma(a) \cdot \Gamma(2-b)} \right\}.$$

Вырожденная гипергеометрическая функция $F(a, b, x)$ вычисляется разложением в ряд

$$F(a, b, x) = 1 + \sum_{i=1}^N \left[\prod_{j=0}^{i-1} \frac{(a+j)}{(1+j)(b+j)} x \right] + \theta_{N+1},$$

дающем при $N=22$ погрешность $\delta < 1 \cdot 10^{-4}$ для $0 \leq a \leq b$ и $-5 \leq x \leq 10$.

Программа 6.21.

Пример. $F(1; 1; 0,5) = 1,648721721$ при точном значении $1,648721721$.

С помощью гипергеометрических функций можно найти большинство элементарных и специальных функций [36, 42]. Например, функция $F(a, a, x)$ при $b=a$ дает значение e^x . Так, в приведенном примере при $b=a \neq 1$ и $x=0,5$ получаем $F(1; 1; 0,5) = e^{0,5} = 1,648721721$. Аналогичный результат дает вычисление $F(3; 3; 0,5)$ при $a=b=3$.

§ 6.11. Дилогарифм

Дилогарифм — интегральная функция

$$f(x) = - \int_1^x \frac{\ln t}{t-1} dt, \text{ может вычисляться с}$$

помощью численного интегрирования методом Гаусса (программа 4.42). Задав $A=1, B=x$ и $M=8$ для $x=0,4$, получим $f(x) = 0,7275863075$ при точном значении $0,727586308$ и $t_c \approx 10$ с.

Дилогарифм может вычисляться и по разложению в ряд:

$$f(x) = \sum_{i=1}^{\infty} (-1)^i \frac{(x-1)^i}{i^2}, \quad 0 \leq x \leq 2.$$

Программа 6.22.

```

05 PRINT 'ВЫЧИСЛЕНИЕ ДИЛОГАРИФМА'
10 INPUT 'ВВЕДИТЕ X=':X: LET Y=X-1
20 LET I=0: LET S=0: LET R=1: LET P=X
30 LET I=I+1: LET R=R: LET P=P*Y
40 LET I=R*P/I^2: LET S=S+D
50 IF ABS(D)>1E-9 THEN 30
60 PRINT 'ДИЛОГАРИФМ F(X)=':S
70 GOTO 10: END

```

Пример. Для $x=0,1$ получим $f(x) = 1,299715$.

§ 6.12. Функции Кельвина

Функции Кельвина относятся к классу функций Бесселя и являются решениями дифференциального уравнения

$$x^2 \frac{d^2 \omega}{dx^2} + x \frac{d\omega}{dx} - (jx^2 + \nu^2) \omega = 0,$$

записываемыми в виде

$$\omega = \text{ber}_\nu(x) + j \text{bei}_\nu(x).$$

Функции Кельвина вычисляются разложением их в следующие ряды [36]:

$$\text{ber}_\nu(x) =$$

$$= \left(\frac{x}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{\cos[(3\nu/4 + k/2)\pi]}{k! \Gamma(\nu + k + 1)} \left(\frac{x^2}{4}\right)^k,$$

$$\text{bei}_\nu(x) =$$

$$= \left(\frac{x}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{\sin[(3\nu/4 + k/2)\pi]}{k! \Gamma(\nu + k + 1)} \left(\frac{x^2}{4}\right)^k.$$

Для практики интерес представляют в основном функции Кельвина целого порядка $\nu = n$ ($n=0, 1, 2, \dots$), в частности функции $\text{ber } x$ и $\text{bei } x$ ($n=0$). При целом $\nu = n$ имеем $\Gamma(\nu + k + 1) = (n+k)!$, что упрощает вычисления, поскольку отпадает необходимость определять гамма-функцию $\Gamma(\nu + k + 1)$ для каждого члена ряда.

Программа 6.23.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ КЕЛЬВИНА BER(X), BEI(X)'
20 INPUT 'ВВЕДИТЕ 0 ДЛЯ ВЫЧИСЛЕНИЯ BER(X) И 1 ДЛЯ BEI(X) ?':C
30 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ E=':E
40 INPUT 'ЗАДАЙТЕ U, X 'U, X: LET A=(X/2)^U: LET B=X**X/4: LET I=0: LET P=1
45 LET I=I+1: LET P=P*I: IF I<U THEN 45
50 LET R=1/P: LET Q=(3*U/4)**PI: IF C=0 THEN 60
55 LET Z=SIN(Q): GOTO 70
60 LET Z=COS(Q)
70 LET S=R*Z: LET K=0
80 LET K=K+1: LET R=R*B/K/(K+U): LET Q=(3*U/4+K/2)**PI
82 IF Q<2**PI THEN 90
85 LET Q=Q-2**PI: GOTO 82
90 IF C=0 THEN 110
100 LET Z=SIN(Q): GOTO 120
110 LET Z=COS(Q)
120 LET W=S: LET S=S+R*Z
125 IF ABS(Z)<E THEN 80
130 IF ABS(S-W)>E THEN 80
140 LET S=S*A: IF C=1 THEN 160
150 PRINT 'BERU(X)=':S: GOTO 40
160 PRINT 'BEIU(X)=':S: GOTO 40: END

```

Пример. При заданной погрешности $E=1 \cdot 10^{-6}$ получим $\text{ber}_2(1,55) = 0,059944$ и $\text{bei}_2(1,55) = -0,259803$ (все цифры верные).

§ 6.13. Функции Дебая и Зиверта

Интегральная функция Дебая $D_n(x) =$

$$= \int_0^x \frac{t^n dt}{e^t} dt \text{ легко находится численным}$$

интегрированием с помощью программы 4.42. Так, для $A=0, B=x=1$ и $n=N=1$ при $M=4$ получим $D_1(x) = 0,7775046341$ при точном значении $0,777505$, аналогично $D_4(1) = 0,6137185176$ (время счета менее 10 с).

$$\text{Интеграл Зиверта } Ziv(z) = \int_0^{\theta} e^{-z/\cos x} dx$$

также легко находится численным интегрированием. Для $z=1, A=0, B=\theta=\pi/2$ и $M=8$ получим $Ziv(1) = 0,3282894817$ при точном значении $0,328286$ и 16 с.

§ 6.14. Интеграл вероятности и родственные ему функции

$$\text{Интеграл вероятности } \text{erf } x = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

желательно вычислять с высокой точностью. Это возможно с применением разложения $\text{erf } x$ в ряд [36]:

$$\text{erf } x = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{n! (2n+1)}.$$

Дополнительный интеграл вероятности

$$\text{erfc } x = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt = 1 - \text{erf } x.$$

Программа 6.24.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ ERF(X) И ERF(X) '
20 INPUT 'ВВЕДИТЕ X='X:LETR=1:LETS=X:LETN=0
30 LETN=N+1:LETK=2*N+1:LETR=-R/N
40 LETW=S:LETS=S+(X^K)*R/K
50 IF ABS(W-S)>=1E-10 THEN 30
60 LETE=2*S/SQR(PI):PRINT'ERF(X)='E
70 LETF=1-E:PRINT'ERFC(X)='F:GOTO 20:END
    
```

В программе 6.24 вычисления заканчиваются, когда очередной член ряда по абсолютной величине становится меньше $\epsilon = 1 \cdot 10^{-10}$.

После этого находятся связанные с ней функции $Q(x)$ и $A(x)$.

```

05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ Z(X), P(X), Q(X) И A(X) '
10 PRINT ' С ПОМОЩЬЮ РАЗЛОЖЕНИЯ P(X) В РЯД '
20 INPUT 'ВВЕДИТЕ X='X:LETR=EXP(-X*X/2)/SQR(2*PI)
30 PRINT'Z(X)='Z:LETR=1:LETS=0:LETN=0
40 LETK=2*N+1:LETR=R*K:LETW=S
50 LETS=S+(X^K)/R:LETN=N+1
60 IF S-W>= 1E-10 THEN 40
70 LETP=.5+S*Z:PRINT'P(X)='P
80 LETQ=1-P:PRINT'Q(X)='Q
90 LETA=2*P-1:PRINT'A(X)='A:GOTO 20:END
    
```

Пример. $\text{erf}(1) = 0,842700793$; $\text{erfc}(1) = 0,1572992071$ ($t_c \approx 5$ с), $\text{erf}(2) = 0,995322265$ и $\text{erfc}(2) = 0,04677734976$ ($t_c \approx 12$ с).

Интеграл вероятности $\text{erf} x$ находится также численным интегрированием. Так, используя программу 4.42 и вводя $A=0$, $B=x=0,5$ и $M=4$, получим $\text{erf} x = 0,5204998779$ при $t_c \approx 6$ с (неверен лишь последний знак).

Гауссовское (нормальное) распределение вероятности характеризуется следующими основными функциями [36]:

$$Z(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, P(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt,$$

$$Q(x) = 1 - P(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt,$$

$$A(x) = 2P(x) - 1 = \frac{1}{\sqrt{2\pi}} \int_{-x}^x e^{-t^2/2} dt.$$

Функцию $P(x)$ удобно вычислять разложением в следующий ряд:

$$P(x) = \frac{1}{2} + Z(x) \sum_{n=0}^{\infty} \frac{x^{2n+1}}{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n+1)}.$$

Пример. $Z(1) = 0,2419707275$, $P(1) = 0,8413447461$, $Q(1) = 0,158655239$, $A(1) = 0,6826894921$ ($t_c \approx 4$ с), $Z(5) = 1,486719515 \times 10^{-6}$, $P(5) = 0,9999997133$, $Q(5) = 2,86686 \times 10^{-7}$ и $A(5) = 0,9999994266$ ($t_c \approx 20$ с).

Для уменьшения времени счета функции $\text{erf} x$ и $P(x)$ могут вычисляться с помощью полиномиальной аппроксимации. Для функции $\text{erf} x$ с погрешностью не выше $1,5 \cdot 10^{-7}$ справедлива аппроксимация

$$\text{erf} x = 1 - (a_1 t + a_2 t^2 + \dots + a_5 t^5) e^{-x^2},$$

где $t = 1/(1 + px)$, $p = 0,3275911$, $a_1 = 0,254829592$, $a_2 = -0,284496736$, $a_3 = 1,421413741$, $a_4 = -1,453152027$ и $a_5 = 1,061405429$.

С погрешностью менее $7,5 \cdot 10^{-8}$ $P(x)$ вычисляется по формуле

$$P(x) = 1 - Z(x) (b_1 t + b_2 t^2 + \dots + b_5 t^5),$$

где $t = 1/(1 + px)$, $p = 0,2316419$, $b_1 = 0,31938153$, $b_2 = -0,356563782$, $b_3 = 1,781477937$, $b_4 = -1,821255978$ и $b_5 = 1,330274429$.

С помощью программы 6.26 с применением этих аппроксимаций вычисляются следующие функции: $P(X) \equiv Z(x)$, $\Phi(X) \equiv \text{erf} x$, $\Pi(X) \equiv P(x)$, $Q(X) \equiv Q(x)$ и $A(X) \equiv A(x)$.

Программа 6.26.

```

05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ P(X), Ф(X), П(X), Q(X) И A(X) '
10 INPUT 'ВВЕДИТЕ X='X: LETY=1/(1+.3275911*X)
20 LETZ=((1.0614054*Y-1.453152)*Y+1.4214137)*Y
30 LETZ=((Z-.28449673)*Y+.2548295)*Y/EXP(X^2)
40 LETF=1-Z: LETY=1/(1+.2316419*X)
50 LETP=EXP(-X^2/2)/SQR(2*PI)
60 LETZ=((1.3302744*Y-1.821256)*Y+1.7814779)*Y
70 LETZ=((Z-.35656378)*Y+.31938153)*Y*P
80 PRINT!1,7!'P(X)='P:PRINT'Ф(X)='F:LETR=1-Z
90 PRINT'П(X)='Z:PRINT'Q(X)='1-Z
100 PRINT'A(X)='2*Z-1:GOTO 10:END
    
```

Пример. $P(1) = 0,2419707245$, $\Phi(1) = 0,8413447497$, $\Pi(1) = 0,8413447497$, $Q(1) = 0,1586552503$ и $A(1) = 0,6826894994$.

§ 6.15. Некоторые статистические функции

К основным статистическим функциям относятся плотности вероятности и функции распределения для различных законов распределения. В большинстве своем эти функ-

Таким образом, для реализации этих вычислений необходимо вычисление гамма-функции $\Gamma(v/2)$. Поскольку указанные выше функции обычно вычисляются при целых $v=1, 2, \dots$, $\Gamma(A) = \Gamma(v/2)$ можно вычислять по следующему алгоритму.

1. Если $v=1$, $A=1/2$ и $\Gamma(A) = \sqrt{\pi}$.
2. Если v — четное число, то $\Gamma(A) = (A-1)!$
3. Если v — нечетное число, то $\Gamma(A) = (A-1)(A-2)\dots(1/2)\sqrt{\pi}$.

Программа 6.27.

```

05 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ X^2 РАСПРЕДЕЛЕНИЯ'
10 INPUT 'ВВЕДИТЕ X^2, U 'X, U: LET A=U/2: LET B=X/2: LET C=A
20 LET G=1: IF INT(A)=0 THEN 40
30 FOR I=1 TO A-1: LET G=G*I: NEXT I: GOTO 55
40 LET G=SQR(#PI): IF C<1 THEN 55
50 LET D=B*C: LET C=C-1: GOTO 40: LET G=6*D/X
55 LET F=(X^(A-1))/G/EXP(A)/(2^A)
60 PRINT 'ЗНАЧЕНИЕ F(X^2, U)='F
70 LET S=0: LET K=0: LET W=0: LET Z=0: LET G=1
80 LET K=K+1: LET W=W+2: LET Z=S: LET G=6*(U+W)
90 LET S=S+(X^K)/G: IF S<Z THEN 80
100 LET P=2*W*F*(1+S)/U: LET Q=1-P
110 PRINT 'ЗНАЧЕНИЕ P(X^2, U)='P
120 PRINT 'ЗНАЧЕНИЕ Q(X^2, U)='Q: GOTO 10: END

```

ции выражаются через элементарные и специальные функции включая последние для нормального распределения (§ 6.14). В первом случае вычисление основных статистических функций не представляет труда, а во втором требует применения некоторых специальных приемов.

Распределение x^2 характеризуется плотностью вероятности

$$f(x) = F(X) = x^{v/2-1} / (2^{v/2} \Gamma(v/2)) e^{-x/2}$$

и функциями распределения ($x = x^2$)

$$P(x, v) = \frac{1}{2^{v/2} \Gamma(v/2)} \int_0^x t^{v/2-1} e^{-t/2} dt,$$

$$Q(x, v) = 1 - P(x/v) =$$

$$= \frac{1}{2^{v/2} \Gamma(v, 2)} \int_0^\infty t^{v/2-1} e^{-t/2} dt.$$

Вычисление $P(x/v)$ выполняется разложением в ряд:

$$P(x/v) = \frac{2x}{v} f(x) \left[1 + \sum_{k=1}^{\infty} \frac{x^k}{(v+2)(v+4)\dots(v+2k)} \right].$$

Пример. $F(1,1) = 0,2419707245$, $P(1,1) = 0,6826894921$, $Q(1,1) = 0,05121793333$ ($t \approx 4$ с) и $F(30,30) = 0,05121793333$, $P(30,30) = 0,534346291$, $Q(30,30) = 0,465553709$ ($t \approx 20$ с).

t -распределение характеризуется функцией

$$I(x, v) = \int_{-x}^x \frac{\Gamma\left(\frac{v+1}{2}\right) \left(1 + \frac{t^2}{v}\right)^{-(v+1)/2}}{\sqrt{\pi v} \Gamma(v/2)} dt.$$

Для v четных функция $I(x, v)$ вычисляется разложением в ряд:

$$I(x, v) = \sin \theta \left[1 + \frac{1}{2} \cos^2 \theta + \frac{1 \cdot 3}{2 \cdot 4} \cos^4 \theta + \dots + \frac{1 \cdot 3 \cdot 5 \dots (v-3)}{2 \cdot 4 \cdot 6 \dots (v-2)} \cos^{v-2} \theta \right],$$

а для v нечетных (кроме $v=1$, когда $I(x, v) = 20/\pi$)

$$I(x, v) = \frac{2\theta}{\pi} + \frac{2}{\pi} \cos \theta \left\{ \sin \theta \left[1 + \frac{2}{3} \cos^2 \theta + \dots + \frac{2 \cdot 4 \dots (v-3)}{1 \cdot 3 \dots (v-2)} \cos^{v-3} \theta \right] \right\},$$

где $\theta = \arctg(x/\sqrt{v})$.

Программа 6.28.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ФУНКЦИЙ I(X, U) T-РАСПРЕДЕЛЕНИЯ'
20 INPUT 'ВВЕДИТЕ X, U 'X, U: LET B=1: LET F=1: LET A=1: LET E=0
30 LET X=ATN(X/SQR(U)): LET C=COS(X)^2
40 IF U/2-INT(U/2)=0 THEN 90
50 LET B=2*X/#PI: IF U=6 THEN 140
60 LET G=U-1: LET D=3
70 LET E=E+2: IF E=5 THEN 150

```

```

80 LET F=F*E/I*С:LETA=A+F:LETD=D+2:GOTO 70
90 LET G=V:LETD=1
100 LET E=E+2:IF E=6 THEN 130
110 LET F=F*D/E*С
120 LET A=A+F:LETD=D+2:GOTO 100
130 LET Z=SIN(X)*A:GOTO 160
140 LET Z=B:GOTO 160
150 LET Z=2/PI*СOS(X)*SIN(X)*A+B
160 PRINT'ЗНАЧЕНИЕ I(X,U)='Z:GOTO 20:END

```

Пример. Для $x=2,13$ и $v=10$ получим $I(x, v) = 0,940978732$.

Функция распределения $Q(x)$ для F -распределения

$$Q(x) =$$

$$= \int_x^{\infty} \frac{\Gamma\left(\frac{v_1+v_2}{2}\right) y^{(v_1/2-1)} (v_1/v_2)^{(v_1/2)} dy}{\Gamma\left(\frac{v_1}{2}\right) \Gamma\left(\frac{v_2}{2}\right) \left(1 + \frac{v_1}{v_2} y\right)^{(v_1+v_2)/2}}$$

при целых v_1 вычисляется разложением в ряд

$$Q(x) = t^{v_2/2} \left[1 + \frac{v_2}{2} (1-t) + \frac{v_2(v_2+2)}{2 \cdot 4} \times \right. \\ \times (1-t)^2 + \dots + \frac{v_2(v_2+2) \dots (v_2+v_1-4)}{2 \cdot 4 \dots (v_1-2)} \times \\ \left. \times (1-t)^{(v_1-2)/2} \right],$$

а при целых v_2

$$Q(x) = 1 - (1-t)^{v_1/2} \left[1 + \frac{v_1}{2} t + \frac{v_1(v_1+2)}{2 \cdot 4} \times \right. \\ \left. \times t^2 + \dots + \frac{v_1(v_1+2) \dots (v_1+v_2-4)}{2 \cdot 4 \dots (v_2-2)} t^{(v_2-2)/2} \right],$$

где $t = v_2 / (v_2 + v_1 x)$.

Если целыми являются v_1 и v_2 , берется ряд для меньшего из них.

Программа 6.29.

```

10 PRINT'ВЫЧИСЛЕНИЕ ФУНКЦИИ Q(X) F-РАСПРЕДЕЛЕНИЯ'
20 INPUT'ВВЕДИТЕ X,U1,U2 'X,U,V,C
30 LET G=C/(C+X*B):LETA=1:LETJ=1:LETK=1:LETK=1:LETI=0
40 LET F=C/2:IF B/2-INT(B/2)=0 THEN 70
50 IF F-INT(F)=0 THEN 130
60 LET Z=0:GOTO 220
70 IF F-INT(F)=0 THEN 90
80 GOTO 100
90 IF B-C=0 THEN 130
100 LET X=0:LETF=0:LETE=C:LETH=1-6
110 IF(B-2)/2=K THEN 190
120 LETK=(B-2)/2:GOTO 160
130 LETX=1:LETF=0:LETE=B:LETH=6
140 IF(C-2)/2=K THEN 190
150 LETK=(C-2)/2
160 LETI=I+1:IF I=K THEN 190
170 LET F=F+2:LETJ=J*E/F*H:LETA=A+J
180 LET E=E+2:GOTO 160
190 LET A=A+J*E/(F+2)*H:IF X=0 THEN 210
200 LET H=1-6:LETD=1-A*H^(B/2):GOTO 220
210 LET Z=A*G^(C/2)
220 PRINT'Q(X,U1,U2)='Z:GOTO 20:END

```

Пример. Для $x=2,71$, $v_1=6$ и $v_2=7$ получим $Q(x) = 0,1090775999$, а для $x=4,12$, $v_1=4$ и $v_2=10$ получим

$$Q(x) = 0,03158204352.$$

В ряде случаев необходимо найти значение аргумента по заданному значению функции вероятности. Общим методом решения этой задачи является решение нелинейного уравнения $F(x) - F_0 = 0$, где $F(x)$ — функция и F_0 — ее заданное значение. Однако значительное сокращение времени вычислений достигается, если использовать аппроксимацию для зависимости x от F_0 . Например, значение $x = x_p$ для заданной функции $Q(x)$ с погрешностью менее $4,5 \cdot 10^{-4}$ вычисляется с помощью следующей аппроксимации [36], справедливой для нормального распределения:

$$x_p = t - \frac{a_0 + a_1 t + a_2 t^2}{1 + b_1 t + b_2 t^2 + b_3 t^3},$$

где

$$t = \sqrt{\ln\left(\frac{1}{p}\right)},$$

$$P = Q(x_p),$$

$$a_0 = 2,515517,$$

$a_1 = 0,802853$, $a_2 = 0,010328$, $b_1 = 1,432788$, $b_2 = 0,189269$ и $b_3 = 0,001308$. Если задана функция $P(x_p) = p_0$, то вычисляется $Q(x_p) = 1 - p_0$, после чего находится x_p .

Программа 6.30.

```
10 PRINT 'ВЫЧИСЛЕНИЕ XР ПО ЗАДАННОМУ ЗНАЧЕНИЮ P=P(XP)'  
20 INPUT 'ВВЕДИТЕ P>=1/2 P='P:LETQ=1-P:LETT=SOR(LOG(1/Q/Q))  
30 LETA=(.010328*T+.802853)*T+2.515517  
40 LETB=((.001308*T+.189269)*T+1.432788)*T  
50 LETX=T-A/(1+B):PRINT!1.4!'XP='X:GOTO 20:END
```

Пример. Для $P(x) = 0,84134$ получим $x_p = 0,999$ (точное значение 1), а для $P(x) = 0,53983$ получим $x_p = 0,0998$ (точное значение 0,1).

Через описанные выше специальные функции могут быть выражены и многие другие статистические функции для различных законов распределения вероятностей. На основе приведенных выше программ могут быть составлены подпрограммы для вычисления специальных функций. Для современных ПЭВМ с большим объемом ОЗУ (32 Кбайт и выше) можно составить единую библиотеку подпро-

грамм специальных функций, что значительно расширяет функциональные возможности ПЭВМ в выполнении сложных математических и статистических расчетов. Для ПЭВМ с графическими возможностями с помощью описанных программ можно организовать вычисление координат точек, необходимых для построения графиков функций.

Следует отметить, что приведенные выше расчеты не исчерпывают всего разнообразия вычислительных задач, связанных с вычислением специальных функций, применяемых в различных сферах науки и техники.

ПРИКЛАДНЫЕ ПРОГРАММЫ ТЕХНИЧЕСКИХ И ЭКОНОМИЧЕСКИХ РАСЧЕТОВ

§ 7.1. Типовые электротехнические расчеты

Преобразование звезды в треугольник (рис. 7.1) и наоборот выполняется по формулам

$$Y \rightarrow \Delta$$

$$R_1 = R/R_5, \quad R_2 = R/R_6, \quad R_3 = R/R_4,$$

$$R = R_4R_5 + R_5R_6 + R_6R_4.$$

$$\Delta \rightarrow Y$$

$$R_4 = R_1R_2/R, \quad R_5 = R_2R_3/R,$$

$$R_6 = R_3R_1/R; \quad R = R_1 + R_2 + R_3.$$

Программа 7.1.

Пример. Для $R_1 = 12$ Ом, $R_2 = 47$ Ом и $R_3 = 82$ Ом (треугольник) получим $R_4 = 4$ Ом, $R_5 = 27,333$ Ом и $R_6 = 6,979$ Ом (звезда). Для $R_4 = 100$ Ом, $R_5 = 150$ Ом и $R_6 = 220$ Ом (звезда) получим $R_1 = 466,667$ Ом, $R_2 = 318,182$ Ом и $R_3 = 700$ Ом (треугольник).

Расчет сопротивлений аттенуаторов (рис. 7.2) при равных активных сопротивлениях генератора $Z_0 = Z$ и нагрузки $Z_1 = Z$ по заданному ослаблению A (в непах):

Расчет ослабления A (в децибелах) Т- и П-образных аттенуаторов при произвольных сопротивлениях Z_0 генератора и Z_1

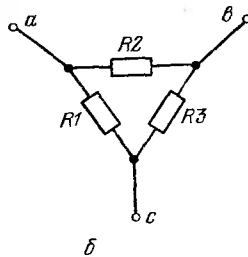
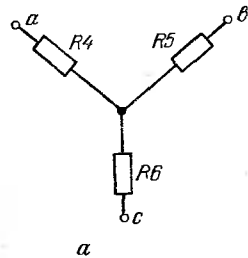


Рис. 7.1. Звезда (а) и треугольник (б), составленные из резисторов

Тип аттенуатора

1	2	3	4
$R = Z \operatorname{sh} A$ $2W = Z/\operatorname{th}(A/2)$	$R/2 = Z \operatorname{th}(A/2)$ $W = Z/\operatorname{sh} A$	$R = Z \operatorname{th}(A/2)$ $W = Z/\operatorname{th}(A/2)$	$K = e^A - 1$

Программа 7.2.

Пример. Для $Z = 100$ Ом и $A = 1$ неп получим

нагрузки выполняется по формулам

$$A_T = 20 \lg \left(\frac{((1 + AR_1/R_2)BR_3 + AR_1)/C}{ZK} \right),$$

$$A_{\Pi} = 20 \lg \left(\frac{((1 + AR_2/Z_0)BZ_0 + AZ_1)/C}{Z/K} \right),$$

где $A = 1 + Z_0/R_1$, $B = 1 + Z_1/R_3$, $C = Z_0 + Z_1$.

Тип аттенуатора

1	2	3	4
$R = 117,520$ Ом $2W = 216,395$ Ом	$R/2 = 46,212$ Ом $W = 85,092$ Ом	$R = 46,212$ Ом $W = 216,395$ Ом	$K = 1,718$ $ZK = 171,818$ Ом $Z/K = 58,198$ Ом

Программа 7.1.

```

10 PRINT 'ПРЕОБРАЗОВАНИЕ ТРЕУГОЛЬНИКА В ЗВЕЗДУ-КОД 0'
20 PRINT 'ПРЕОБРАЗОВАНИЕ ЗВЕЗДЫ В ТРЕУГОЛЬНИК-КОД 1'
30 INPUT 'ВВЕДИТЕ КОД ПРЕОБРАЗОВАНИЯ ? ' K: IF K=1 THEN 80
40 INPUT 'ВВЕДИТЕ R1, R2, R3 ' R1, R2, R3
50 LET R=R1+R2+R3: LET R4=R1*R2/R
60 LET R5=R2*R3/R: LET R6=R3*R1/R
70 PRINT 'R4=' R4' R5=' R5' R6=' R6: GOTO 30
80 INPUT 'ВВЕДИТЕ R4, R5, R6 ' R4, R5, R6
90 LET R=R4*R5+R5*R6+R6*R4
100 LET R1=R/R5: LET R2=R/R6: LET R3=R/R4
110 PRINT 'R1=' R1' R2=' R2' R3=' R3: GOTO 30: END
    
```

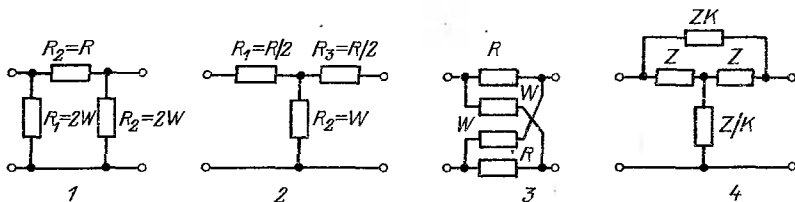


Рис. 7.2. Резистивные аттенюаторы четырех типов

Программа 7.2.

```

10 PRINT 'РАСЧЕТ СОПРОТИВЛЕНИЙ АТТЕНЮАТОРОВ'
15 PRINT '5.31' ПО ЗАДАННОМУ ОСЛАБЛЕНИЮ А'
20 INPUT 'ВВЕДИТЕ ВХОДНОЕ И ВЫХОДНОЕ СОПРОТИВЛЕНИЯ Z=' Z
25 INPUT 'ВВЕДИТЕ ОСЛАБЛЕНИЕ В ДБ В ПЕРВАХ А=' A: LET B=A/2
30 INPUT 'ВВЕДИТЕ ТИП АТТЕНЮАТОРА 1, 2, 3 ИЛИ 4 ?' C
40 IF C=1 THEN 80
50 IF C=2 THEN 90
60 IF C=3 THEN 100
70 PRINT 'АТТЕНЮАТОР 4 ТИПА'
75 PRINT 'R=' Z*HTN(B)' OM W=' Z/HTN(B)' OM: GOTO 20
80 PRINT 'АТТЕНЮАТОР 1 ТИПА'
85 PRINT 'R=' Z*HSN(A)' OM Z*W=' Z/HTN(B)' OM: GOTO 20
90 PRINT 'АТТЕНЮАТОР 2 ТИПА'
95 PRINT 'R/2=' Z*HTN(B)' OM W=' Z/HSN(A)' OM: GOTO 20
100 PRINT 'АТТЕНЮАТОР 3 ТИПА': LET K=EXP(A)-1
110 PRINT 'K=' K' Z*K=' Z*K' OM Z/K=' Z/K' OM: GOTO 20: END
    
```

Программа 7.3.

```

10 PRINT 'РАСЧЕТ ОСЛАБЛЕНИЯ Т- И П-ОБРАЗНЫХ АТТЕНЮАТОРОВ'
20 INPUT 'ЗАДАЙТЕ В ОМАХ Z0, Z1 ' K, L
30 INPUT 'ЗАДАЙТЕ В ОМАХ R1, R2, R3 ' H, I, J
40 LET A=1+K/H: LET B=1+L/J: LET C=K+L
50 LET F=20*LG((J*(1+A*H/I)*B+A*H)/C)
60 LET G=20*LG((K*(1+A*I/K)*B+A*I*L)/C)
70 PRINT 'ОСЛАБЛЕНИЕ АТ=' F' ДБ'
80 PRINT 'ОСЛАБЛЕНИЕ АП=' G' ДБ: GOTO 20: END
    
```

Пример. Для $Z_0=Z_1=50$ Ом, $R_1=75$ Ом, $R_2=38$ Ом и $R_3=19$ Ом получим $A_T=11,469$ дБ, $A_P=14,560$ дБ.

Расчет сопротивлений согласующего делителя (рис. 7.3) и вносимого им ослабления выполняется по формулам

$$R_1 = Z_0 \sqrt{1 - Z_1/Z_0}, \quad R_2 = Z_1 / \sqrt{1 - Z_1/Z_0},$$

$$L_{\text{мин}} = 20 \log (\sqrt{Z_0/Z_1} + \sqrt{Z_0/Z_1 - 1}).$$

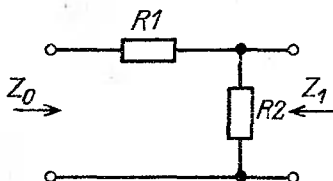


Рис. 7.3. Согласующий резистивный делитель

Программа 7.4.

```

10 PRINT'РАСЧЕТ СОПРОТИВЛЕНИЯ И ОСЛАБЛЕНИЯ РЕЗИСТОРНОГО ДЕЛИТЕЛЯ'
20 INPUT'ВВЕДИТЕ В ОМЫХ СОПРОТИВЛЕНИЯ Z0,Z1 'Z0,Z1:LETK=Z1/Z0
30 LETR1=Z0*SQR(1-K):LETR2=Z1/SQR(1-K)
40 LETL=20*LG1(SQR(1/K)+SQR(1/K-1)):PRINT!5.3!'R1='R1' OM'
50 PRINT'R2='R2' OM':PRINT'ОСЛАБЛЕНИЕ L MIN='L' ДБ':GOTO 20:END
    
```

Пример. Для $Z_0=500$ Ом и $Z_1=200$ Ом получаем $R_1=387,298$ Ом, $R_2=258,199$ Ом и $L_{\text{мин}}=8,961$ дБ.

Расчет сопротивлений Т- и П-образного аттенуаторов по заданному входному Z_0 и выходному Z_1 сопротивлениям и коэффициенту ослабления мощности N выполняется с помощью следующих выражений:

Т-образный аттенуатор

$$R_1 = Z_0 \left(\frac{N+1}{N-1} \right) - R_3,$$

$$R_2 = Z_1 \left(\frac{N+1}{N-1} \right) - R_3,$$

где $R_3 = \frac{2}{N-1} \sqrt{N Z_0 Z_1}$,

П-образный аттенуатор

$$\frac{1}{R_1} = \frac{1}{Z_0} \left(\frac{N+1}{N-1} \right) - \frac{1}{R_3},$$

$$\frac{1}{R_2} = \frac{1}{Z_1} \left(\frac{N+1}{N-1} \right) - \frac{1}{R_3},$$

$$\frac{1}{R_3} = \frac{2}{N-1} \sqrt{\frac{N}{Z_0 Z_1}}.$$

Программа 7.5.

```

10 PRINT'РАСЧЕТ Т- И П-ОБРАЗНЫХ АТТЕНУАТОРОВ'
20 PRINT'ПО ЗАДАННОМУ ОСЛАБЛЕНИЮ МОЩНОСТИ N'
30 INPUT'ВВЕДИТЕ В ОМЫХ СОПРОТИВЛЕНИЯ Z0,Z1 'Z0,Z1
40 INPUT'ВО СКОЛЬКО РАЗ (N) ОСЛАБЛЯЕТСЯ МОЩНОСТЬ N=? 'N
50 LETK=(N+1)/(N-1)
60 INPUT'ЗАДАЙТЕ КОД 0-Т И 1-П ? 'C
70 IF C=1 THEN 100
80 LETR3=2*SQR(N*Z0*Z1)/(N-1)
90 LETR1=Z0*K-R3:LETR2=Z1*K-R3:GOTO 130
100 LETG3=2*SQR(N/Z0/Z1)/(N-1):LETR3=1/G3
110 LETG1=K/Z0-G3:LETG2=K/Z1-G3
120 LETR1=1/G1:LETR2=1/G2
130 PRINT!5.3!'R1='R1' R2='R2' R3='R3':GOTO 40:END
    
```

Пример. При $Z_0=200$ Ом, $Z_1=100$ Ом и $N=50$ для Т-образного аттенуатора находим $R_1=167,347$ Ом, $R_2=63,265$ Ом и $R_3=40,816$ Ом, а при $Z_0=100$ Ом, $Z_1=$

$=50$ Ом и $N=100$ для П-образного аттенуатора $R_1=136,147$ Ом, $R_2=56,990$ Ом и $R_3=350,018$ Ом.

Сопротивление последовательной (рис. 7.4, а) и параллельной (рис. 7.4, б) RCL цепей рассчитывается по формулам ($\omega = 2\pi f$, где f — частота).

Цепь последовательная

$$Z = R + j\omega L = X + jY,$$

$$|Z| = \sqrt{R^2 + (\omega L - 1/\omega C)^2},$$

$$\theta = \arctg \left(\frac{\omega L - 1/\omega C}{R} \right).$$

Цепь параллельная

$$Z = \frac{\frac{1}{R} - j(\omega C - 1/\omega L)}{\frac{1}{R^2} + (\omega C - 1/\omega L)^2} = X + jY.$$

$$|Z| = \sqrt{1/R^2 + (\omega C - 1/\omega L)^2},$$

$$\theta = \arctg \left[R \left(\frac{1}{\omega L} - \omega C \right) \right],$$

где $|Z|$ — модуль полного (комплексного) сопротивления, θ — угол сдвига фаз, X — действительная и Y — мнимая составляющие Z .

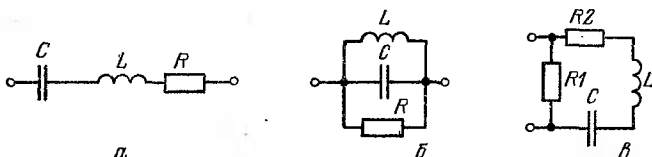


Рис. 7.4. Последовательная (а) и параллельная (б) LCR цепи и колебательный контур (в)

Программа 7.6.

```

10 PRINT 'СОПРОТИВЛЕНИЕ LCR ЦЕПЕЙ'
20 PRINT 'ЗАДАЙТЕ КОД 0-ЕСЛИ ЦЕПЬ ПОСЛЕДОВАТЕЛЬНАЯ'
30 INPUT 'И КОД 1-ЕСЛИ ЦЕПЬ ПАРАЛЛЕЛЬНАЯ ? ' K
40 INPUT 'ВВЕДИТЕ В ОМАХ СОПРОТИВЛЕНИЕ R=' R
50 INPUT 'ВВЕДИТЕ В МИКРОФАРАДАХ ЕМКОСТЬ C=' C: LETC=C/1E6
60 INPUT 'ВВЕДИТЕ В МИЛЛИГЕНРИ ИНДУКТИВНОСТЬ L=' L: LETL=L/1E3
65 INPUT 'ЗАДАЙТЕ В ГЕРЦАХ ЧАСТОТУ F=' F
70 LETW=2*#PI*F: IF K=1 THEN 100
80 LETX=R: LETY=(W*L-1/W/C): LETG=ATN(Y/X)
90 LETZ=SGR(X*X+Y*Y): GOTO 130
100 LETA=W*C-1/W/L: LETB=1/R/A*#A
110 LETX=1/R/B: LETY=-A/B: LETZ=1/SGR(B)
120 LETG=ATN(A*B)
130 PRINT 'КОМПЛЕКСНОЕ СОПРОТИВЛЕНИЕ Z=' X' + J*(Y)'
140 PRINT 'МОДУЛЬ /Z/=' Z' Ом'
150 PRINT 'ФАЗОВЫЙ УГОЛ G=' G*180/#PI' ГРАДУСОВ'
160 GOTO 65: END
    
```

Пример. Для последовательной RCL цепи при $R=10$ Ом, $C=5$ мкФ, $L=20$ мГн и $f=60$ Гц получим $X=10$ Ом, $Y=-522,977$ Ом, $|Z|=523,072$ Ом и $\theta=G=-88,905^\circ$. Для параллельной RCL цепи при $R=47$ Ом, $C=1$ мкФ, $L=30$ мГн и $f=50$ Гц получаем $X=1,827$ Ом, $Y=9,085$ Ом, $|Z|=9,267$ Ом и $\theta=G=78,628^\circ$.

Сопротивление резонансного контура (рис. 7.4, в) Z и его резонансная частота f_0 вычисляются по формулам ($\omega=2\pi f$)

$$f_0 = 1/2\pi \sqrt{LC},$$

$$Z = X + jY =$$

$$= \frac{R_1 R_2 (R_1 + R_2) + R_1 (\omega L - 1/\omega C)^2}{(R_1 + R_2)^2 + (\omega L - 1/\omega C)^2} +$$

$$+ j \frac{R_1^2 (\omega L - 1/\omega C)}{(R_1 + R_2)^2 + (\omega L - 1/\omega C)^2},$$

$$|Z| = \sqrt{X^2 + Y^2}, \quad \theta = \arctg(Y/X).$$

Программа 7.7.

```

10 PRINT 'ВЫЧИСЛЕНИЕ СОПРОТИВЛЕНИЯ РЕЗОНАНСНОГО КОНТУРА'
15 INPUT 'ВВЕДИТЕ В ОМАХ СОПРОТИВЛЕНИЯ R1,R2 ' R1,R2
20 INPUT 'ВВЕДИТЕ В МИЛЛИГЕНРИ ИНДУКТИВНОСТЬ L=' L: LETL=L/1E3
30 INPUT 'ВВЕДИТЕ В МИКРОФАРАДАХ ЕМКОСТЬ C=' C: LETC=C/1E6
40 INPUT 'ЗАДАЙТЕ В ГЕРЦАХ ЧАСТОТУ F=' F
50 LETW=2*#PI*F
60 LETW=W*L-1/W/C: LETT=(R+S)^2+W^2
70 LETX=(R*S*(R+S)+R*W*W)/T
80 LETY=R*R*W/T: LETZ=SGR(X*X+Y*Y): LETS=ATN(Y/X)
90 PRINT 'РЕЗОНАНСНАЯ ЧАСТОТА F0=' 1/2/#PI/SGR(L*#C)
100 PRINT 'КОМПЛЕКСНОЕ СОПРОТИВЛЕНИЕ Z=' X' + J*(Y)'
110 PRINT 'МОДУЛЬ /Z/=' Z' Ом'
120 PRINT 'ФАЗОВЫЙ УГОЛ Q=' 180*S/#PI: GOTO 40: END
    
```

Пример. Для $R_1=20$ Ом, $R_2=15$ Ом, $C=0,5$ мкФ, $L=20$ мГн, $f=60$ Гц находим $f_0=F_0=1591,549$ Гц, $X=20,000$ Ом, $Y=-0,076$ Ом, $|Z|=20,000$ Ом и $\theta=-0,216^\circ$.

Расчеты сложных цепей выполняются на постоянном токе методами контурных токов и узловых потенциалов. В конечном счете они сводятся к решению линейных или нелинейных систем уравнений.

Пример 1. Определить токи I_1 , I_2 и I_3 в мостовой схеме (рис. 7.5), содержащей три контура. Применяя метод контурных токов, задаем произвольные направления токов и составляем уравнения для контуров. Каждое уравнение базируется на том, что сумма э. д. с. и падений напряжения на резисторах каждого контура равна 0.

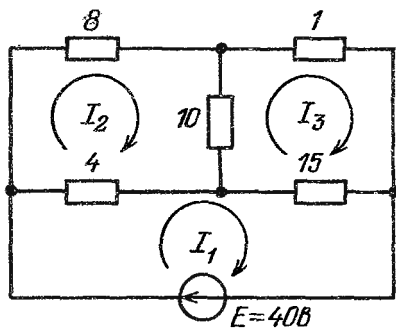


Рис. 7.5. Мостовая схема

Контур	Уравнение контура
1	$4I_1 - 4I_2 + 15I_1 - 15I_3 - 40 = 0$
2	$4I_2 - 4I_1 + 8I_2 + 10I_2 - 10I_3 = 0$
3	$10I_3 - 10I_2 + I_3 + 15I_3 - 15I_1 = 0$

В этих уравнениях коэффициенты перед I_1 , I_2 , I_3 — сопротивления соответствующих резисторов схемы (рис. 4.5).

После простых преобразований из этих уравнений получаем систему линейных уравнений:

$$\begin{aligned} 19I_1 - 4I_2 - 15I_3 &= 40, \\ -4I_1 + 22I_2 - 10I_3 &= 0, \\ -15I_1 + 10I_2 + 26I_3 &= 0. \end{aligned}$$

Решая ее, например, с помощью программы 4.1, получаем: $I_1 = 7,860$ А, $I_2 = 4,230$ А и $I_3 = 6,192$ А.

Пример 2. Определить ток I_k в интегральной схеме токоотвода (стабилизатора тока) (см. рис. 7.6). В общем случае это достаточно сложная задача, требующая применения громоздкой эквивалентной схемы транзистора. Однако нормальная работа токоотвода обеспечивается при работе транзистора в линейном режиме, когда $I_k = BI_2$ и $I_3 = (B+1)I_2$ и напряжение база-эмиттер

$$U_{03} = m\varphi_T \ln \frac{(B+1)I_2}{I_{30}},$$

где $m \approx 2$ — константа, учитывающая реком-

```

10 PRINT 'РАСЧЕТ ТОКОВ В СХЕМЕ ТОКООТВОДА'
20 INPUT 'ВВЕДИТЕ K, FT, E, R1, R2, R3, K, F, E, R, Z, W
30 INPUT 'ВВЕДИТЕ M, I0, I30, B, M, I, J, V
35 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА H='H
40 INPUT 'ЗАДАЙТЕ НАЧАЛЬНЫЕ ТОКИ I10, I20, X, Y: LET B=B+1
50 LET A=M*F: LET D=A*K
60 LET C=D*LOG((X-Y)/I+1)
70 LET X=(E-C)/R: LET T=Y
80 LET Y=(C-A*LOG(B*Y/J)-W*B*Y)/Z
90 IF ABS(Y-T)>H THEN 60
100 PRINT 'ТОК I1='X: PRINT 'ТОК I2='Y
110 PRINT 'НАПРЯЖЕНИЕ НА ДИОДАХ UD='C
120 PRINT 'ТОК КОЛЛЕКТОРА IK='Y*(B-1)
130 END
    
```

бинацию носителей в эмиттерном переходе. $\varphi_T = kT/q$ — тепловой потенциал (k — постоянная Больцмана, T — температура, q — заряд электрона), B — коэффициент передачи тока базы и I_{30} — обратный ток эмиттерного перехода.

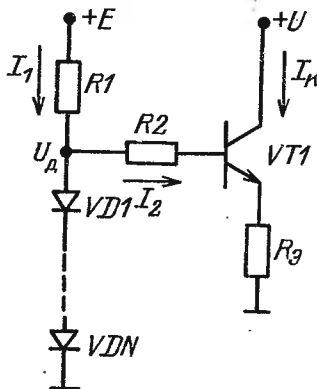


Рис. 7.6. Интегральная схема токоотвода

Падение напряжения на k диодах цепи базового смещения можно найти с помощью

формулы

$$U_d = km\varphi_T \ln \left(\frac{I_1 - I_2}{I_0} + 1 \right),$$

где I_0 — обратный ток диодов.

Итак, применяя метод узловых потенциалов, получаем для токов I_1 и I_2 следующую систему нелинейных уравнений:

$$I_{1n+1} = \frac{E - U_{dn}}{R_1},$$

$$I_{2n+1} = \frac{U_{dn} - m\varphi_T \ln \frac{(B+1)I_{2n}}{I_{30}} - R_3(B+1)I_{2n}}{R_2},$$

где $U_{dn+1} = km\varphi_T \ln \left(\frac{I_{1n} - I_{2n}}{I_0} + 1 \right)$. Эта си-

стема непосредственно пригодна для решения ее методом простых итераций (см. § 4.4). Учитывая самостоятельное значение рассчитываемого U_d , составим для решения этой системы отдельную программу.

Программа 7.8.

Пример. Для $k=2$, $\varphi_T=0,025$ В, $E=5$ В, $R_1=2$ кОм, $R_2=20$ кОм, $R_3=0,1$ кОм, $m=2$, $I_0=1 \cdot 10^{-6}$ мА, $I_{30}=1 \cdot 10^{-7}$ мА, $B=50$, погрешности $H=1 \cdot 10^{-6}$, начального значения $I_{20}=0,01$ мА, $I_{10}=1$ мА получаем: $I_1=1,781$ мА, $I_2=0,02472$ мА, $U_d=1,4379$ В и $I_k=1,2358$ мА.

§ 7.2. Расчет индуктивных элементов

Расчет индуктивных элементов сводится к расчету индуктивности L элемента заданной конструкции или какого-либо параметра последней при заданной индуктивности. Для последнего числа индуктивных элементов существуют достаточно простые формулы для практических расчетов индуктивности L . Приводим для справок основные из них с контрольными примерами (L в наногенри, размеры в сантиметрах) [22].

1. Индуктивность отрезка круглого провода длиной l и диаметром d

$$L = 2l \left(\operatorname{arsh} \frac{l}{d} + \frac{d}{l} - \sqrt{\left(\frac{d}{l} \right)^2 + 1} \right).$$

Для $l=1$ см и $d=0,1$ см $L=4,186$ нГн.

2. Индуктивность отрезка провода прямоуг-
 ольного сечения (ширина b , толщина t)
 длиной l $L = 2l \left(\ln \frac{2l}{b+t} + 0,447 \frac{t+b}{2l} + \frac{1}{2} \right)$.
 Для $b=0,1$ см, $t=0,001$ см и $l=1$ см $L =$
 $=7,017$ нГн.

3. Индуктивность одиночного круглого
 витка диаметром D и диаметром провода d
 $L = 2\pi D \left(\ln \frac{8D}{d} - 1,75 \right)$. Для $D=2$ см и
 $d=0,1$ см $L = 41,785$ нГн.

4. Индуктивность тонкопленочного круг-
 лого витка диаметром D и шириной полоски
 $b \gg t$ (t — толщина полоски) $L = 2\pi D \times$
 $\times \left(\ln \left(\frac{2\pi D}{b} \right) - 2,45i \right)$. Для $b=0,01$ см и
 $D=1$ см $L = 25,983$ нГн.

5. Индуктивность тонкопленочного квад-
 ратного витка со стороны квадрата l и
 шириной полоски $b \gg t$ $L = 8l \left(\ln \frac{8l}{b} - 2,583 \right)$.
 Для $l=1$ см и $b=0,01$ см $L = 30,653$ нГн.

6. Индуктивность однослойной катушки
 круглого сечения со средним радиусом R и
 длиной катушки l $L = 4\pi\omega^2 R \left[\ln \left(1 + \frac{\pi}{l/R} \right) + \right.$
 $\left. + \frac{1}{2,3 + 1,6l/R + 0,44 (l/R)^2} \right]$. Формула обес-
 печивает погрешность менее 0,1%. Для
 $R=1$ см, $l=2$ см и $\omega=10$ имеем $L =$
 $=1359,627$ нГн.

7. Индуктивность однослойной катушки
 квадратного сечения со средней длиной
 стороны квадрата $2Q$ и длиной всей
 катушки l $L = 16 \cdot \omega^2 a \left[\ln \left(1 + \frac{\pi}{(b/a)} \right) + \right.$
 $\left. + \frac{1}{3,64 + 2 (l/a) + 0,51 (l/a)^2} \right]$. Для $a=1$ см,
 $l=2$ см и $\omega=40$ имеем $L = 26816$ нГн.

8. Индуктивность тороидальной однослой-
 ной катушки с тором круглого сечения со
 средним диаметром тора D и диаметром
 сечения D_1 :

$$L = 2\pi\omega^2 (D - \sqrt{D^2 - D_1^2}). \quad (7.1)$$

Для $D=3$ см, $D_1=1$ см и $\omega=300$ имеем
 $L = 83486$ нГн.

9. Индуктивность тороидальной однослой-
 ной катушки квадратного сечения с внут-
 ренним диаметром тора D_1 , наружным D_2 и
 высотой h

$$L = 2\omega^2 h \ln (D_2/D_1). \quad (7.2)$$

Для $D_2=3$ см, $D_1=1$ см, $h=1$ см и $\omega=100$
 имеем $L = 21972$ нГн.

10. Индуктивность многослойной торо-
 идальной катушки со средним диаметром
 тора D (включая обмотку) и диаметром
 сечения тора D_1

$$L = 2\pi D \omega^2 \left(\ln \frac{8D}{D_1} - 1,75 \right). \quad (7.3)$$

Формула справедлива при плотной намотке
 и при условии, что толщиной изоляции про-

вода можно пренебречь. Для $D=4$ см,
 $D_1=0,8$ см и $\omega=100$ имеем $L = 487293$ нГн.

11. Индуктивность многослойной короткой
 цилиндрической катушки со средним диа-
 метром D , длиной $l < D$ и толщиной обмот-
 ки c $L = \frac{25\pi\omega^2 (D^2)}{3D + 9l + 10c}$.

Для $D=2,5$ см, $l=1$ см, $c=1$ см и
 $\omega=100$ имеем $L = 185235$ нГн.

12. Индуктивность катушки на тороидаль-
 ном ферромагнитном сердечнике прямоуг-
 лого сечения с плотной намоткой

$$L = \frac{4\mu h (d_n - d_b) \omega^2}{(d_n + d_b)} \text{ при } d_n/d_b < 1,5 \div 2, \quad (7.4)$$

$$L = 2\omega^2 \mu h \ln (d_n/d_b) \text{ при } d_n/d_b > 1,5 \div 2, \quad (7.5)$$

где μ — магнитная проницаемость сердеч-
 ника, h — его высота, d_n и d_b — наружный
 и внутренний диаметры кольца. Для $d_n =$
 $=4$ см, $d_b = 2$ см, $h = 1$ см, $\mu = 1000$, $\omega = 120$.
 имеем $L = 19962638$ нГн, а при $d_b = 3$ см и
 $\omega = 200$ (остальные данные прежние) $L =$
 $= 22857142$ нГн.

13. Индуктивность катушек на П- и Ш-
 образных замкнутых ферромагнитных сердеч-
 никах $L = \frac{4\mu F_c \omega^2}{l_m (1 + \mu d_l / a l_m)}$, где F_c — пло-
 щадь поперечного сечения сердечника, d_l —
 толщина зазора, l_m — длина магнитной линии
 сердечника, $a = 1 - 2$ — поправочный коэффи-
 циент на рассеивание в зазоре.

Для $F_c = 1$ см², $d_l = 0,05$ см, $l_m = 10$ см,
 $a = 1,5$, $\mu = 1000$ и $\omega = 400$ имеем $L =$
 $= 46398906$ нГн.

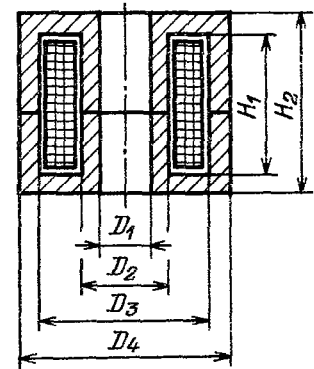


Рис. 7.7. Броневой цилиндрический сердечник

14. Индуктивность катушки в броневом
 цилиндрическом сердечнике (рис. 7.7)

$$L = \frac{19,74\mu\omega^2}{A + B}, \quad (7.6)$$

где

$$A = (h_1 + h_2) \left(\frac{1}{d_2^2 - d_3^2} + \frac{1}{d_2^2 - d_1^2} \right), \quad (7.7)$$

$$B = \frac{1}{h_2 - h_1} \ln \left(\frac{d_3 + d_4}{d_1 + d_2} \right). \quad (7.8)$$

Для $d_1=1$ см, $d_2=2$ см, $d_3=4$ см, $d_4=5$ см, $h_1=3$ см, $h_2=4$ см, $\mu=10$ и $\omega=100$ имеем $L=468914$ нГн.

15. Индуктивность тонкопленочных катушек $L=k_1(A_n+A_{вн})\omega^{5/3}\ln\left(k_2\frac{A_n+A_{вн}}{A_n-A_{вн}}\right)$, где $k_1=2,33$ и $k_2=4$ для спиральной (круглой) катушки, $k_1=12,05$ и $k_2=8$ — для квадратной, A_n и $A_{вн}$ — наружный и внутренний размеры. Для спиральной катушки с $\omega=5$, $A_n=1$ см и $A_{вн}=0,5$ см получим $L=126,972$ нГн.

16. Взаимная индуктивность двух тонких проводов длиной l , расположенных параллельно на расстоянии h , $M=2l\left(\operatorname{arsh}\frac{l}{h}+\frac{h}{l}-\sqrt{\left(\frac{h}{l}\right)^2+1}\right)$. Для $h=1$ см и

$l=10$ см $L=41,865$ нГн.

17. Взаимная индуктивность двух тонких проводов длиной a и b , расположенных на одной прямой с расстоянием между стыкуемыми концами h , $M=(a+b+h)\ln(a+b+h)+h\ln h-(a+h)\ln(a+h)-(b+h)\ln(b+h)$. Для $a=10$ см, $b=5$ см и $h=1$ см $M=7,234$ нГн.

18. Взаимная индуктивность двух проводов длиной l , сходящихся в одной точке с расстоянием между сходящимися концами и углом между ними φ , $M=2l\cos\varphi\ln(1+2l/c)$.

Для $l=20$ см, $c=5$ см и $\varphi=30^\circ$ получим $M=76,114$ нГн.

19. Взаимная индуктивность двух тонких круглых катушек с диаметром D и средним расстоянием между ними H $M=0,6\omega_1\omega_2D^4/H^3$.

```

10 PRINT 'ЧИСЛО ВИТКОВ ТОРОИДАЛЬНОЙ КАТУШКИ'
20 PRINT 'ОДНОСЛОЙНОЙ, КРУГЛОГО СЕЧЕНИЯ КОД 1'
30 PRINT 'ОДНОСЛОЙНОЙ, ПРЯМОУГОЛЬНОГО СЕЧЕНИЯ КОД 2'
40 PRINT 'МНОГОСЛОЙНОЙ, КРУГЛОГО СЕЧЕНИЯ КОД 3'
45 INPUT 'ВВЕДИТЕ КОД' N:IF N=1 THEN 70
50 IF N=3 THEN 70
60 INPUT 'ВВЕДИТЕ ВЫСОТУ СЕЧЕНИЯ H='H
70 INPUT 'ВВЕДИТЕ ДИАМЕТР D='D
80 INPUT 'ВВЕДИТЕ ДИАМЕТР X='X
90 INPUT 'ВВЕДИТЕ ИНДУКТИВНОСТЬ L='L
100 IF N=1 THEN 140
110 IF N=2 THEN 160
120 IF N=3 THEN 180
130 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 45
140 LET W=SQR(L/(2*#PI*(D-SQR(D^2-X^2))))
150 GOTO 190
160 LET W=SQR(L/(2*#PI*LOG(D/X)))
170 GOTO 190
180 LET W=SQR(L/(2*#PI*D*(LOG(8*D/X)-1,75)))
190 PRINT 'ЧИСЛО ВИТКОВ W='W
200 GOTO 90 : END

```

где ω_1 и ω_2 — число витков катушек. Для $\omega_1=\omega_2=50$, $D=2$ см и $H=0,5$ см $M=192000$ нГн.

20. Взаимная индуктивность двух коаксиальных катушек, намотанных на общий тороидальный сердечник (материал не ферромагнитный), $M=4l\omega_1\omega_2F_{ср}/l$, где $F_{ср}$ — площадь среднего поперечного сечения (с учетом толщины обмотки первой катушки, запол-

няющей весь сердечник), l — длина средней окружности тора.

Однократные расчеты по приведенным формулам целесообразно выполнять в калькуляторном режиме.

Пример 3. Определить взаимную индуктивность катушек (см. 20), если $\omega_1=200$, $\omega_2=50$, $F_{ср}=1$ см² и $l=8$ см. Обозначив $\omega_1=W_1$, $\omega_2=W_2$, $F_{ср}=F$ и $l=H$, выполняем следующие операции:

W1=200	ПС	(ввод ω_1)
W2=50	ПС	(ввод ω_2)
F=1	ПС	(ввод $F_{ср}$)
H=8	ПС	(ввод l)

PRINT 4*#PI*W1*W2*F/H ПС
Получаем значение $L=15707,96327$ нГн.

При многократно повторяющихся вычислениях и при громоздких формулах расчеты L , ω или других параметров катушек целесообразно выполнять по программам. Поскольку составление программ формульного счета достаточно просто, ограничимся приведением нескольких характерных примеров.

Пример 4. Составить программу для расчета числа витков, необходимых для получения заданной индуктивности L для трех типов тороидальных катушек: 1 — однослойной круглого сечения, 2 — однослойной прямоугольного сечения и 3 — многослойной круглого сечения. Необходимо использовать формулы (7.1) — (7.3), разрешив их относительно ω , и предусмотреть выбор типа катушки и расчет по соответствующей для нее формуле. Это можно сделать вводом кода, указывающего на тип катушки, и использованием в программе условных переходов для выбора нужной формулы.

Программа 7.9.

Пример.

Код 1	Код 2
D=3 см	h=H=1 см
D1=X=1 см	D=3 см
	D1=X=1 см
L=1·10 ⁵ нГн	L=1·10 ⁵ нГн
$\omega=W=304,569$	$\omega=W=213,335$

Код 3
 $D=4$ см
 $D1=X=0,8$ см

$L=1 \cdot 10^7$ нГн
 $\omega=W=453,007$

В ряде случаев зависимость L от ω или от другого параметра x неразрешима относительно искомого параметра x при заданном L_0 . В этом случае значение x может быть получено из решения нелинейного уравнения

$$L_0 - L(x) = 0 \quad (7.12)$$

подходящим численным методом. Удобным является метод поразрядного приближения, при котором не требуется анализа сходимости и приближение $L(x)$ к значению L_0

Пример 5. Составить программу для расчета числа витков катушки на тороидальном сердечнике, используя при $d_n/d_b < 1,75$ формулу (7.4), а при $d_n/d_b \geq 1,75$ — формулу (7.5).

Программа 7.10.

```
10 PRINT'РАСЧЕТ L ИЛИ W КАТУШКИ НА ТОРОИДАЛЬНОМ'
20 PRINT' ФЕРРОМАГНИТНОМ СЕРДЕЧНИКЕ'
30 PRINT'НЕИЗВЕСТНОМУ L ИЛИ W ЗАДАЙТЕ НУЛЕВОЕ ЗНАЧЕНИЕ'
35 INPUT'ВВЕДИТЕ ВНЕШНИЙ И ВНУТРЕННИЙ ДИАМЕТРЫ ТОРА D, D1 'D, I
40 INPUT'ВВЕДИТЕ ВЫСОТУ ТОРА H='H
50 INPUT'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ МАГНИТНУЮ ПРОНИЦАЕМОСТЬ M='M
60 INPUT'ЗАДАЙТЕ ИНДУКТИВНОСТЬ КАТУШКИ L='L
70 INPUT'ЗАДАЙТЕ ЧИСЛО ВИТКОВ КАТУШКИ W='W
80 LETK=D/I:LETA=0:IF K<1.75 THEN LETA=1
90 IF A=0 THEN 110
100 LETF=(D+I)/4/M/H/(D-I):GOTO 120
110 LETF=1/2/M/H/LOG(D/I)
120 IF W=0 THEN 140
130 PRINT'ИНДУКТИВНОСТЬ L='W*W/F:GOTO 60
140 PRINT'ЧИСЛО ВИТКОВ W='SQR(L*F):GOTO 60:END
```

Для $d_n=D=4$ см, $d_b=D1=2$ см, $h=H=1$ см, $\mu=M=1000$, $L=2 \cdot 10^7$ нГн получим $\omega=W=120,112$, а для $D=4$ см, $D1=3$ см, $H=1$ см, $M=1000$ и $L=2 \cdot 10^7$ нГн получим $W=187,083$.

Пример 6. Составить программу для расчета L по заданному ω или ω по заданному L для катушки на броневом сердечнике. Для этого формулу (7.6) представим в виде

$$L = \omega^2 / F, \quad (7.9)$$

$$\omega = \sqrt{LF}, \quad (7.10)$$

$$F = (A + B) / (19,74\mu), \quad (7.11)$$

где A и B определяются выражениями (7.7) и (7.8). Для вычисления L (по заданному ω) или ω (по заданному L) удобно неизвестному параметру задавать при вводе нулевое значение и с помощью оператора условного перехода задавать вычисления по нужной формуле — (7.9) или (7.10).

Программа 7.11.

```
10 PRINT'РАСЧЕТ L ИЛИ W КАТУШКИ НА БРОНЕВОМ'
20 PRINT' ФЕРРОМАГНИТНОМ СЕРДЕЧНИКЕ'
30 INPUT'ВВЕДИТЕ ДИАМЕТРЫ D1, D2, D3, D4 'D, P, Q, R
40 INPUT'ВВЕДИТЕ РАЗМЕРЫ H1, H2 'H, I
50 INPUT'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ МАГНИТНУЮ ПРОНИЦАЕМОСТЬ M='M
60 LEA=(H+I)*(1/(R*R-Q*Q))+1/(P*P-O*O)
70 LETB=LOG((R+Q)/(P+O))/(I-H):LETF=(A+B)/19.74/M
80 PRINT'НЕИЗВЕСТНОМУ L ИЛИ W ЗАДАЙТЕ НУЛЕВОЕ ЗНАЧЕНИЕ'
90 INPUT'ВВЕДИТЕ L, W 'L, W:IF W=0 THEN 110
100 PRINT'ИНДУКТИВНОСТЬ L='W*W/F:GOTO 80
110 PRINT'ЧИСЛО ВИТКОВ W='SQR(L*F):GOTO 80:END
```

Для $D1=1$ см, $D2=2$ см, $D3=4$ см, $D4=5$ см, $H1=3$ см, $H2=4$ см, $\mu=10$, $L=5 \cdot 10^{-5}$ нГн, задав $W=0$, получим $W=103,261$, а задав $W=100$ и $L=0$, получим $L=468914$ нГн.

происходит при увеличении x с 0. Последнее исключает необходимость в задании начального интервала изменения $L(x)$, который заданно неизвестен.

Пример 7. По технологическим соображениям внутренний размер $A_{вн}=A$ и шаг витков Z тонкопленочных катушек должны быть заданными величинами. Найти число витков тонкопленочной катушки для этого случая при заданной индуктивности L_0 .

Зависимость $L(\omega)$ в этом случае имеет вид

$$L(\omega) = k_1 \left(1 + \frac{2A}{\omega Z} \right) Z \omega^{2/3} + \ln \left[k_2 \left(1 + \frac{2A}{\omega Z} \right) \right].$$

Приведенная ниже программа решает уравнение (7.12) при $x=\omega$ методом поразрядного приближения и выдает результат с заданной погрешностью решения $\epsilon=Q$ (это, однако, не означает, что погрешность расче-

та ω также равна ϵ , — последняя прежде всего зависит от погрешности исходной формулы для L , которая может быть существенно выше ϵ).

Программа 7.12.

```

10 PRINT 'ЧИСЛО ВИТКОВ ТОНКОПЛЕНОЧНОЙ КАТУШКИ С ЗАДАНЫМ ШАГОМ'
20 INPUT 'КРУГЛОЙ-КОД 1, КВАДРАТНОЙ-КОД 2. ВВЕДИТЕ КОД 'N
50 IF N=1 THEN 70
60 GOTO 80
70 LET K1=2.33 : LET K2=4 : GOTO 110
80 IF N<>2 THEN 100
90 LET K1=12.05 : LET K2=8 : GOTO 110
100 PRINT 'КОД НАБРАН НЕВЕРНО': GOTO 20
110 INPUT 'ВВЕДИТЕ ВНУТРЕННИЙ РАЗМЕР A='A
120 INPUT 'ВВЕДИТЕ ШАГ Z='Z
130 INPUT 'ЗАДАЙТЕ ТОЧНОСТЬ РЕЗУЛЬТАТА Q='Q
140 INPUT 'ВВЕДИТЕ ИНДУКТИВНОСТЬ L='L: LET W=0: LET H=1
150 IF ABS(H)<Q THEN 200
170 LET W=W+H: GOSUB 210: IF F/H>=0 THEN 170
190 LET H=-H/10 : GOTO 160
200 PRINT 'ЧИСЛО ВИТКОВ W='W : GOTO 140
210 LET V=(2*A/(W*Z))+1: LET F=L-K1*V*Z*(W^(8/3))*LOG(K2*V)
220 RETURN : END

```

Для квадратной катушки (код 2) с $A=0,2$ см, $Z=0,005$ см, $L=100$ нГн и $\epsilon=1 \cdot 10^{-3}$ получим $W=6,935$.

Программная реализация расчетов индуктивных элементов целесообразна также, если расчеты проводятся по многим сопряженным формулам. Ниже даны два таких примера.

Пример 8. Расчет дросселя фильтра на стальном О-, П- или Ш-образном сердечнике. задается индуктивность дросселя L и рабочий ток обмотки I .

1. Определяется ширина стержня $a=2,6\sqrt{LI^2}$, на котором находится обмотка, и сечение его $Q=1,5a^2$.

2. Выбирается стандартный сердечник со значениями a и Q , равными полученным или несколько большими их. Для выбранного сердечника известны ширина окна b , высота окна h и длина магнитной линии l_m .

3. Вычисляем толщину набора $c=Q/a$.

4. Вычисляем коэффициент $k=Ll^2/acl_m$.

5. Используя аппроксимацию зависимости оптимальной толщины воздушного зазора $l_z\%$ (в % от l_m) от k , находим $l_z\%$ по формуле $l_z\%=(330k-15 \cdot 10^3k^3)$.

Аналогично эффективную магнитную проницаемость материала сердечника определяем так: $M=\mu_z=(114 \exp(-401k)+50)$.

6. Находим половину длины оптимального зазора $l_z/2=l_z\%l_m/20$.

7. Задав допустимую плотность тока в проводе обмотки, находим диаметр провода $d=1,13\sqrt{I_0/d}$.

8. Определяем число витков обмотки $\omega=10\sqrt{LI_m/1,26\mu_zac}$.

9. Вычисляем коэффициент заполнения окна сердечника $k_n=0,008\omega d^2/bh$.

10. Находим омическое сопротивление обмотки дросселя $r=0,000225\omega \cdot 2(a+c)+\pi b/d^2$.

Программа 7.13.

```

10 PRINT 'РАСЧЕТ ДРОССЕЛЯ ФИЛЬТРА'
20 INPUT 'ВВЕДИТЕ ИНДУКТИВНОСТЬ L='L
30 INPUT 'ВВЕДИТЕ ТОК I='I
40 LET A=2.6*((L*(I^2))^(1/4))
45 PRINT 'ШИРИНА СТЕРЖНЯ A='A
50 LET Q=(A^2)*1.5
55 PRINT 'СЕЧЕНИЕ СЕРДЕЧНИКА Q='Q
60 PRINT 'ВЫБИРАТЕ СТАНДАРТНЫЙ СЕРДЕЧНИК ПО ЗНАЧЕНИЯМ A И Q'
70 INPUT 'ВВЕДИТЕ ШИРИНУ СТЕРЖНЯ A И ОКНА B' A, B
80 INPUT 'ВВЕДИТЕ ВЫСОТУ ОКНА H='H
90 INPUT 'ВВЕДИТЕ ДЛИНУ МАГНИТНОЙ ЛИНИИ L1='L1
95 INPUT 'ВВЕДИТЕ СЕЧЕНИЕ СЕРДЕЧНИКА Q='Q
100 LET C=Q/A
105 PRINT 'ТОЛЩИНА НАБОРА C='C
110 LET K=(L*(I^2))/(A*C*L1)
115 PRINT 'ВСПОМОГАТЕЛЬНЫЙ КОЭФФИЦИЕНТ K='K
120 LET X=330*K-15000*(K^3)
125 PRINT 'ОПТИМАЛЬНАЯ ДЛИНА ВОЗДУШНОГО ЗАЗОРА X='X
130 LET M=114*EXP(-401*K)+50
135 PRINT 'ЭФФЕКТИВНАЯ МАГНИТНАЯ ПРОНИЦАЕМОСТЬ M='M
140 LET Y=X*L1/20
150 INPUT 'ВВЕДИТЕ ДОПУСТИМУЮ ПЛОТНОСТЬ ТОКА В ПРОВОДЕ P='P
160 LET D=SQR(I/P)*1.13
165 PRINT 'ДИАМЕТР ПРОВОДА D='D
170 LET W=10000*SQR(L*L1/(1.26*M*A*C))
180 PRINT 'ЧИСЛО ВИТКОВ ДРОССЕЛЯ W='W
190 LET K1=0.008*W*(I^2)/(B*H)
200 PRINT 'КОЭФФИЦИЕНТ ЗАПОЛНЕНИЯ ОКНА СЕРДЕЧНИКА ДРОССЕЛЯ K1='K1
210 LET R=0.000225*W*(2*(A+C)+#PI*B)/(I^2)
220 PRINT 'ОМИЧЕСКОЕ СОПРОТИВЛЕНИЕ ОБМОТКИ ДРОССЕЛЯ R='R
230 GOTO 10 : END

```

Для $L = 3 \cdot 10^{-4}$ Гн и $I = 15$ А получаем $a = A = 1,325$ см и $Q = 2,634$ см². Выбираем из стандартного ряда сердечник, имеющий $a = 1,6$ см, $Q = 2,65$ см, $b = 1,6$ см, $h = 4$ см, $l_m = 13,7$ см. Введя эти данные, получаем $C = 1,656$ см, $\mu_z = 104,089$ и $l_z/2 = 0,385$ см. Далее, введя $\delta = 3$ А/мм², получаем $d = 2,527$ мм, $\omega = 34,388$, $k_m = 0,274$ и $r = 0,014$ Ом.

Пример 9. Расчет силового трансформатора, работающего от промышленной сети переменного тока (частота 50 или 60 Гц).

10. Определим число витков первичной обмотки $\omega = (U_1 - \Delta U_1) / I_1$.

Если трансформатор имеет несколько вторичных обмоток, то их отдаваемые в нагрузку мощности суммируются, так что

$$P_{\text{гав}} = I_1 U_1 = \frac{1}{\eta_{\text{тр}}} \sum_{i=2}^N I_i U_i, \text{ где } N - \text{общее число}$$

обмоток. При этом расчет ω_i и d_i ($i \geq 2$) повторяется для каждой обмотки.

Программа 7.14.

```

10 PRINT'РАСЧЕТ СИЛОВОГО ТРАНСФОРМАТОРА'
20 INPUT'ВВЕДИТЕ ГАБАРИТНУЮ МОЩНОСТЬ P='P
30 INPUT'ВВЕДИТЕ МАКСИМАЛЬНУЮ ИНДУКЦИЮ В СЕРДЕЧНИКЕ I='I
40 INPUT'ВВЕДИТЕ ЧИСЛО СТЕРЖНЕЙ СЕРДЕЧНИКА С ОБМОТКАМИ S='S
50 INPUT'ВВЕДИТЕ ДОПУСТИМУЮ ПЛОТНОСТЬ ТОКА ПРОВОДОВ P='X
60 INPUT'ВВЕДИТЕ КОЭФФ. ЗАПОЛНЕНИЯ ОКНА СЕРДЕЧНИКА МЕДЬЮ KM='K1
70 INPUT'ВВЕДИТЕ КОЭФФ. ЗАПОЛНЕНИЯ СЕРДЕЧНИКА СТАЛЬЮ KC='K2
80 INPUT'ЗАДАЙТЕ К.П.Д. ТРАНСФОРМАТОРА N='N
90 INPUT'ЗАДАЙТЕ ЧАСТОТУ СЕТИ F='F
100 INPUT'ВВЕДИТЕ ДЛЯ ПЕРВИЧНОЙ ОБМОТКИ U1,I1 'U1,I1
110 INPUT'ВВЕДИТЕ ДЛЯ ВТОРИЧНОЙ ОБМОТКИ U2,I2 'U2,I2
120 LETY=P/F/S/I*X/N/K1/K2/.0222:PRINT'6.3!'QST*Q0='Y
130 PRINT'ПОДБЕРИТЕ ПОДХОДЯЩИЙ ТИП СЕРДЕЧНИКА И ДЛЯ НЕГО'
140 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ A,B,C 'A,B,C
150 LETE=F*X*A*K*I*K2*44E-6:PRINT'Э.Д.С ОДНОГО ВИТКА E='E
160 LETW2=U2/E:LETW1=U1/E:LETD2=1.13*SQR(I2/X)
170 LETD1=1.13*SQR(I1/X):LETL1=W1*(2*(A+C)+#PI*B)/100
180 LETU=2.25E-2*I1*L1/(D1^2):LETM=(U1-U)/E
190 PRINT'ДИАМЕТР ПРОВОДА ПЕРВИЧНОЙ ОБМОТКИ D1='D1
200 PRINT'ДИАМЕТР ПРОВОДА ВТОРИЧНОЙ ОБМОТКИ D2='D2
210 PRINT'ЧИСЛО ВИТКОВ ПЕРВИЧНОЙ ОБМОТКИ W1='W1
220 PRINT'ЧИСЛО ВИТКОВ ВТОРИЧНОЙ ОБМОТКИ W2='W2:END

```

При таких частотах несущественны индуктивности рассеивания и расчет может проводиться по следующей методике.

1. Задаем габаритную мощность трансформатора $P_{\text{гав}}$, максимальную индукцию в сердечнике B_m , частоту сети f , число стержней сердечника с обмотками s , допустимую плотность тока δ , к. п. д. трансформатора $\eta_{\text{тр}}$, коэффициент заполнения окна медью k_m , коэффициент заполнения сердечника сталью k_c , напряжение U_1 и U_2 , первичную и вторичную обмотки, токи I_1 и I_2 этих обмоток.

2. Находим произведение сечений стали $Q_{\text{ст}}$ и окна Q_0 : $Q_{\text{ст}} Q_0 = P_{\text{гав}} / (0,0222 f B_m \delta \eta_{\text{тр}} s k_c k_m)$.

3. По значению $Q_{\text{ст}} Q_0$ выбираем подходящий тип магнитопровода, уточняем значения $Q_{\text{ст}}$ и Q_0 и определяем для выбранного сердечника ширину стержня a , ширину окна b и толщину набора c .

4. Находим э. д. с., наводимую в одном витке $e = 4,44 f B_m Q_{\text{ст}} k_c \cdot 10^{-4}$.

5. Вычисляем число витков вторичной обмотки $\omega_2 = U_2 / e$.

6. Находим первое приближение для числа витков первичной обмотки $\omega_1 = U_1 / e$.

7. Находим диаметр проводов первичной и вторичной обмоток $d_1 = 1,13 \sqrt{I_1 / \delta}$, $d_2 = 1,13 \sqrt{I_2 / \delta}$.

8. Вычисляем длину провода первичной обмотки $l_1 = \omega_1 [2(a+c) + \pi b] \cdot 10^{-2}$.

9. Вычисляем падение напряжения на омическом сопротивлении первичной обмотки $\Delta U_1 = 2,25 \cdot 10^{-2} I_1 l_1 / d_1^2$.

Для $P_{\text{гав}} = 215$ Вт, $B_m = 1,2$ Тл, $S = 2$, $F = 50$ Гц, $\delta = 2$ А/мм², $k_c = 0,95$, $k_m = 0,32$, $\eta_{\text{тр}} = 0,96$, $U_1 = 127$ В, $U_2 = 450$ В, $I_1 = 1,7$ А и $I_2 = 0,475$ А находим $Q_{\text{ст}} Q_0 = 138,270$ см². Подходящим является ленточный магнитопровод (сердечник) ПЛ25×50—60 с сечением стержня 25×50 мм и размерами окна 40×65 мм ($a = 2,5$ см, $b = 2$ см и $c = 5$ см). У этого сердечника $Q_{\text{ст}} Q_0 = 162$ см⁴. Введя эти данные, получим $d_2 = 0,551$ мм², $d_1 = 1,042$ мм, $\omega_2 = 1422,475$ и $\omega_1 = 391,396$.

§ 7.3. Расчет емкостных элементов и конденсаторов

Для емкостных элементов известны выражения, определяющие емкость (далее в пикофарадах) как функцию их геометрических размеров (в сантиметрах) и относительной диэлектрической проницаемости диэлектрика ϵ_d . Ниже приведены формулы для емкости наиболее распространенных элементов (конденсаторов) [22].

1. Плоский конденсатор в виде пластины диэлектрика с нанесенными с двух сторон металлическими обкладками: $C = 0,0886 \epsilon_d S / h$, где S — площадь обкладок, h — толщина диэлектрика. Для $S = 1$ см², $h = 0,1$ см, $\epsilon_d = 1$ (воздух) имеем $C = 0,886$ пФ. Формула не учитывает краевые эффекты.

2. Дисковый конденсатор с электродами одинакового размера, занимающими всю

$$C=0,0695\epsilon_d D/h. \quad (7.13)$$

Для $D=1$ см, $h=0,1$ см и $\epsilon_d=4,7$ получим $C=3,266$ пФ.

3. Коаксиальный (трубчатый) конденсатор с электродами и диэлектриком одинаковой длины l : $C=0,241\epsilon_d l/\lg(D_n/D_v)$, где D_n — наружный и D_v — внутренний диаметры трубки. Для $l=2$ см, $D_n=0,5$ см, $D_v=0,4$ см и $\epsilon_d=4,7$ $C=23,376$ пФ. Формула не учитывает краевые эффекты.

4. Коаксиальный (трубчатый) конденсатор с электродами одинаковой длины l и диэлектриком, длина трубки которого больше l : $C=\frac{0,241\epsilon_d l + 0,08D_n}{\lg(D_n/D_v)}$. Для $l=2$ см, $D_n=0,5$ см, $D_v=0,4$ см и $\epsilon_d=4,7$ получим $C=23,789$ пФ.

5. Коаксиальный (трубчатый) конденсатор с электродами разной длины (меньшая — l): $C=\frac{0,241\epsilon_d (l + 0,17D_n)}{\lg(D_n/D_v)}$. Для $\epsilon_d=4,7$, $l=2$ см, $D_n=0,5$ см и $D_v=0,4$ см получим $C=24,370$ пФ.

6. Шаровой конденсатор: $C=0,555\epsilon_d D_n/(D_n/D_v - 1)$, где D_n и D_v — внешний и внутренний диаметры шара. Для $D_n=10$ см, $D_v=9$ см и $\epsilon_d=4,7$ получим $C=234,765$ пФ.

8. Шар в свободном пространстве, заполненном диэлектриком: $C=0,555\epsilon_d D$, где D — диаметр шара. Для $D=10$ см и $\epsilon_d=1$ получим $C=5,55$ пФ.

9. Емкость между двумя одинаковыми шарами диаметром D и расстоянием между центрами $a > D$:

$$C=0,278\epsilon_d D \left[1 + \frac{D(4a^2 - D^2)}{2a(4a^2 - 2aD - D^2)} \right]. \quad (7.14)$$

Для $D=10$ см, $a=15$ см и $\epsilon_d=1$ получим $C=4,263$ пФ.

10. Круглый диск в свободном пространстве: $C=0,353\epsilon_d D \left(1 + \frac{0,637h}{D} \right)$, где D — диаметр и h — толщина диска. Для $D=10$ см, $h=1$ см, $\epsilon_d=1$ получим $C=3,755$ пФ.

11. Емкости проводников в свободном пространстве: $C=k_1\epsilon_d l/\lg(k_2x/d)$, где значения коэффициентов k_1 и k_2 и расшифровка параметра x даны в табл. 7.1, l — длина проводника и d — его диаметр.

Пример 10. Составить программу для вычисления емкости проводников, указанных в табл. 7.1. Выбор варианта расчета можно обеспечить вводом кода и применением команд условных переходов.

Программа 7.15.

```

10 PRINT 'ЕМКОСТЬ ПРОВОДНИКОВ В СВОБОДНОМ ПРОСТРАНСТВЕ'
20 PRINT 'ЕМКОСТЬ МЕЖДУ ДВУМЯ ПРОВОДАМИ ВАДАИ ОТ ЗЕМЛИ, КОД 1'
30 PRINT 'МЕЖДУ ГОРИЗОНТАЛЬНЫМ ПРОВОДОМ И ЗЕМЛЕЙ, КОД 2'
40 PRINT 'МЕЖДУ ВЕРТИКАЛЬНЫМ ПРОВОДОМ И ЗЕМЛЕЙ ПРИ N>L/4, КОД 3'
50 PRINT 'МЕЖДУ ВЕРТИКАЛЬНЫМ ПРОВОДОМ И ЗЕМЛЕЙ ПРИ N<L/4, КОД 4'
60 PRINT 'МЕЖДУ ПРОВОДОМ И КОРПУСОМ, КОД 5'
70 INPUT 'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ ДИЭЛЕКТРИЧЕСКУЮ ПРОВОДИМОСТЬ E='E
80 INPUT 'ВВЕДИТЕ ДИАМЕТР ПРОВОДА D='D
90 INPUT 'ВВЕДИТЕ КОД' N
100 IF N=1 THEN 150
110 IF N=2 THEN 160
120 IF N=3 THEN 170
130 IF N=4 THEN 180
140 IF N=5 THEN 190
150 LET K1=0.12 : LET K2=2 : GOTO 210
160 LET K1=0.24 : LET K2=4 : GOTO 215
170 LET K1=0.24 : LET K2=1 : GOTO 230
180 LET K1=0.24 : LET K2=1.15 : GOTO 230
190 LET K1=0.24 : LET K2=2 : GOTO 205
200 INPUT 'КОД НАБРАН НЕВЕРНО' : GOTO 90
205 INPUT 'ВВЕДИТЕ ДИАМЕТР ОТВЕРСТИЯ X='X : GOTO 220
210 INPUT 'ВВЕДИТЕ РАССТОЯНИЕ МЕЖДУ ПРОВОДАМИ X='X : GOTO 220
215 INPUT 'ВВЕДИТЕ ВЫСОТУ ПОДВЕСКИ X='X
220 INPUT 'ВВЕДИТЕ ДЛИНУ ПРОВОДА L='L : GOTO 240
230 INPUT 'ВВЕДИТЕ ДЛИНУ ПРОВОДА X='X : GOTO 250
240 LET C=(K1*E*L)/LG(T(K2*(X/D))) : GOTO 260
250 LET C=(K1*E*X)/LG(T(K2*X/D))
260 PRINT 'ЕМКОСТЬ ПРОВОДНИКА В СВОБОДНОМ ПРОСТРАНСТВЕ C='C
270 GOTO 90 : END

```

7. Горшковский конденсатор в виде цилиндра длиной l , закачиваемый снизу полушаром с наружным диаметром D_n и внутренним D_v : $C=\epsilon_d \left[\frac{0,241l}{\lg(D_n/D_v)} + \frac{0,278D_n}{D_n/D_v - 1} \right]$. Для $l=10$ см, $D_n=5$ см, $D_v=4$ см и $\epsilon_d=4,7$ получим $C=143,014$ пФ.

Для $l=10^3$ см, $d=0,2$ см, $x=200$ см и $\epsilon_d=1$ получим $C=66,629$ пФ (код 2).

Более полные данные о расчете емкости можно получить из специальной литературы [13]. Часто необходимо не значение емкости конденсатора, а определение по заданной емкости его электрофизических или геометрических характеристик. Приведем примеры таких расчетов.

Значения k_1 , k_2 и наименование x для расчета емкости проводников в свободном пространстве

Код	Емкость C	k_1	k_2	Наименование x
1	Между двумя проводниками от земли	0,12	2	Расстояние между проводниками
2	Между горизонтальным проводом и землей	0,24	4	Высота подвески
3	Между вертикальным проводом и землей при расстоянии от земли до нижнего конца $h > l/4$	0,24	1	Длина провода
4	Между вертикальным проводом и землей при расстоянии от земли до нижнего конца $h < l/4$	0,24	1,15	Длина провода
5	Между проводом и корпусом при вводе провода через отверстие	0,24	2	Диаметр отверстия

Пример 11. Определить диэлектрическую проницаемость диэлектрика ϵ_d . Для этого удобно использовать дисковый конденсатор специальной конструкции — (см. рис. 7.8). Такой конденсатор имеет охранный кольцо, которое должно иметь потенциал, равный потенциалу верхней обкладки конденсатора. Это резко ослабляет проявление краевых эффектов и позволяет пользоваться формулой (7.13) для нахождения ϵ_d по заданному значению C : $\epsilon_d = Ch/0,0695D$.

Пусть конденсатор с емкостью $C = 32$ пФ имеет $D = 10$ см и $h = 0,1$ см. Тогда диэлектрическая проницаемость диэлектрика $\epsilon_d = 4,604$.

Пример 12. Составить программу для расчета тонкопленочных конденсаторов заданной емкости C . Методика расчета следующая [9].

1. Задаем число обкладок m , толщину диэлектрика d и его диэлектрическую проницаемость ϵ_d .

2. Вычисляем удельную емкость (приходящуюся на 1 см^2) тонкопленочного конденсатора $C_0 = 0,0885\epsilon_d(m-1)/d$ и необходимую площадь обкладок $S = C/C_0$.

3. Задаем отношение сторон $Q = A/B$ прямоугольника обкладок и находим стороны $A = \sqrt{QS}$ и $B = \sqrt{S/Q}$.

Программа 7.16.

```

10 PRINT 'РАСЧЕТ ПЛЕНОЧНЫХ КОНДЕНСАТОРОВ'
20 INPUT 'ВВЕДИТЕ ЕМКОСТЬ C=' C
30 INPUT 'ВВЕДИТЕ ЧИСЛО ОБКЛАДОК M=' M
40 INPUT 'ВВЕДИТЕ ТОЛЩИНУ ДИЭЛЕКТРИКА D=' D
50 INPUT 'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ E=' E
60 LET C0=0.0885*E*(M-1)/D
70 PRINT 'УДЕЛЬНАЯ ЕМКОСТЬ C0=' C0
80 LET S=C/C0
90 PRINT 'ПЛОЩАДЬ ОБКЛАДОК S=' S
100 INPUT 'ВВЕДИТЕ ОТНОШЕНИЕ СТОРОН Q=A/B' Q
110 LET A=SQR(Q*S) : LET B=SQR(S/Q)
120 PRINT 'СТОРОНА ОБКЛАДКИ A=' A
130 PRINT 'СТОРОНА ОБКЛАДКИ B=' B
140 GOTO 20 : END

```

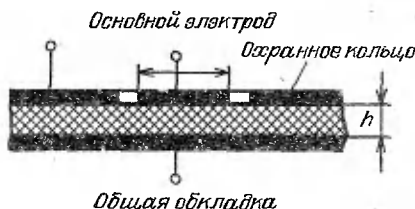


Рис. 7.8. Дисковый конденсатор с охранным кольцом для уменьшения краевых эффектов

Для $m=3$, $d=0,01$ см, $\epsilon_d=4,7$ и $C=100$ пФ получаем $C_0=83,19$ пФ/см², $S=1,202$ см² и при $Q=2$ получаем $A=1,550$ см и $B=0,775$ см.

Пример 13. Найти расстояние a между центрами двух шаров, имеющих диаметр $D=10$ см, при котором емкость между ними составляет $C_0=5$ пФ. Непосредственное применение формулы (7.14) в данном случае невозможно, поскольку ее разрешимость относительно параметра a не является очевидной. В связи с этим целесообразно найти a из решения нелинейного уравнения $C(a) - C_0 = 0$, учитывая, что при $a \rightarrow D$ шары соприкасаются и емкость $C \rightarrow \infty$. Поэтому, применяя метод поразрядного приближения, нужно увеличить a с начального значения

$a=D$. Приведенная ниже программа (в ней $a \equiv X$) реализует такой подход.

Программа 7.17.

```

10 PRINT 'РАССТОЯНИЕ МЕЖДУ ЦЕНТРАМИ ДВУХ ШАРОВ ПО ЗАДАННОЙ ЕМКОСТИ'
20 INPUT 'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ E='E
30 INPUT 'ВВЕДИТЕ ДИАМЕТР ШАРОВ D='D
40 INPUT 'ВВЕДИТЕ ТОЧНОСТЬ РЕЗУЛЬТАТА Q='Q
50 INPUT 'ВВЕДИТЕ ЕМКОСТЬ C='C
60 LET X=D : LET N=10
70 IF ABS(X)<Q THEN 110
80 LET X=X+N : GOSUB 120
90 IF F/N>=0 THEN 80
100 LET N=N/10 : GOTO 70
110 PRINT 'РАССТОЯНИЕ X='X : GOTO 50
120 LET F=.278*E*D*(1+(D*(4**X-D*D)/(2***(4**X-D*D-2**D))))-C
130 RETURN : END

```

Для $C=5$ пФ, $D=10$ см, $\epsilon_d=1$ и погрешности вычислений $Q=1 \cdot 10^{-3}$ получим $a=X=12,26$ см.

ПЭВМ можно использовать и для существенно более сложных вычислений емкости, описанных в [13] и требующих применения специальных методов расчета. Ограничимся одним примером такого рода.

Пример 14. Вычислить емкость на единицу длины C_l между двумя компланарными (т. е. находящимися на одной плоскости параллельно друг другу) пластинами бесконечной длины шириной $a=9$ см и с расстоянием между ними $d=2$ см. В соответствии с [13] эта емкость определяется точной формулой $C_l/\epsilon_d = k(m')/k(m)$, где $m = k^2 = 1/(1+2a/d)^2$, $k(m)$ — полный эллиптический интеграл первого рода, $k(m')$ — полный эллиптический интеграл от параметра $m' = 1-m$. При заданных a и d имеем $m=0,01$ и $m'=0,99$. Воспользовавшись программой 6.18, вычисляем $k(m) = k(0,01) = 1,5745568$ и $k(m') = k(0,99) = 3,695637369$. Следовательно, $C_l = \epsilon_d k(m')/k(m) = \epsilon_d \times 2,346815539$ пФ/см. Это совпадает с приведенным в [13] значением C_l , но исключает необходимость применения таблиц значений эллиптических интегралов и их интерполяции и экстраполяции при произвольных значениях m и m' .

§ 7.4. Расчет линий передачи и задержки

Линии передачи могут быть проводными, коаксиальными (кабельными), ленточными и полосоновыми. Важнейшим параметром таких линий является волновое сопротивление $Z_n = \sqrt{L_1/C_1}$, где L_1 и C_1 — индуктивность и емкость линий на единицу длины, фазовая скорость движения волны в линии $v = c/\sqrt{\epsilon_d}$, где c — скорость света и ϵ_d — относительная диэлектрическая проницаемость диэлектрика линии.

В справочной литературе [22] обычно приводятся формулы для инженерного расчета Z_n линий. Однако в большинстве случаев значение Z_n бывает задано и требуется определить конструктивные параметры линии, обеспечивающие получение нужного значе-

ния Z_n . Ниже рассматривается комплекс программ для таких расчетов.

Для ряда проводных линий обобщенный конструктивный параметр

$$x = \frac{D}{d} = \frac{\exp(Z_n \sqrt{\epsilon_d}/60)}{k}$$

где d — диаметр круглого проводника, а D и k определяются следующим образом.

1. Для линии в виде круглого проводника, расположенного в середине между плоскопараллельными проводящими плоскостями с расстоянием между ними D , $k=1,27$.

2. Для линии в виде проводника, размещенного в центре экрана с квадратным сечением и стороной D , $k=1,08$.

3. Для коаксиальной линии в виде экрана с круглым сечением (диаметр D) и проводником, расположенным в центре, $k=1$.

4. Для линии в виде проводника, расположенного на биссектрисе прямого угла, образованного двумя полубесконечными проводящими плоскостями, на расстоянии $D/2$ от каждой плоскости, $k=1,4$.

В приведенной ниже программе вычисляется параметр любой из этих четырех линий. Тип линии задается кодом.

Программа 7.18.

Для $Z_n=300$ Ом и $\epsilon_d \equiv E=1$ получим $X=116,861$ при коде 1, $X=137,420$ при коде 2, $X=148,413$ при коде 3 и $X=106,009$ при коде 4. Выбрав заданным один из параметров (D или d), по значению X легко найти другой параметр.

Если выражение $Z_n(x)$ неразрешимо относительно требуемого расчетного параметра x , то для нахождения x по заданному $Z_n = Z_{n0}$ решается нелинейное уравнение

$$Z_{n0} - Z_n(x) = 0. \quad (7.15)$$

Например, для линии в виде двух проводников одинакового диаметра D , расположенных на расстоянии A друг от друга и на расстоянии L от проводящей плоскости,

$$Z_n = \frac{120}{\sqrt{\epsilon}} \left[\operatorname{arch} \frac{A}{D} - \ln \sqrt{1 + \left(\frac{A}{2L} \right)^2} \right],$$

а для линии в виде двух проводников различного диаметра (D и X), расположенных

Программа 7.18.

```

10 PRINT 'РАСЧЕТ ПАРАМЕТРА X ПРОВОДЯЩИХ ЛИНИЙ'
40 PRINT 'ЛИНИЯ - КРУГЛЫЙ ПРОВОДНИК, РАСПОЛОЖЕННЫЙ'
50 PRINT 'В СЕРЕДИНЕ МЕЖДУ ПРОВОДЯЩИМИ ПЛОСКОСТЯМИ, КОД 1'
60 PRINT 'В ЭКРАНЕ С КВАДРАТНЫМ СЕЧЕНИЕМ, КОД 2'
70 PRINT 'В ЭКРАНЕ С КРУГЛЫМ СЕЧЕНИЕМ, КОД 3'
80 PRINT 'НА БИСЕКТРИСЕ ПРЯМОГО УГЛА, КОД 4'
90 INPUT 'ВВЕДИТЕ КОД-' N
100 IF N=1 THEN 150
110 IF N=2 THEN 160
120 IF N=3 THEN 170
130 IF N=4 THEN 180
140 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 90
150 LET K=1.27 : GOTO 190
160 LET K=1.08 : GOTO 190
170 LET K=1 : GOTO 190
180 LET K=1.4
190 INPUT 'ВВЕДИТЕ ВОЛНОВОЕ СОПРОТИВЛЕНИЕ ЛИНИИ Z=' Z
195 INPUT 'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ E=' E
200 LET X=(EXP((Z*SQR(E))/60))/K
210 PRINT 'X=' X : GOTO 90 : END

```

на расстоянии A в свободном пространстве,

$$Z_n = \frac{60}{\sqrt{\epsilon}} \operatorname{arch} \frac{4A^2 - D^2 - X^2}{2DX}$$

Если необходимо определить расстояние A по заданному Z_{0n} , то это можно сделать с помощью приведенной ниже программы, обеспечивающей решение (7.15) методом подекадного приближения.

Программа 7.19.

```

10 PRINT 'РАССТОЯНИЕ МЕЖДУ ПРОВОДАМИ ДВУХПРОВОДНОЙ ЛИНИИ'
20 PRINT 'МЕЖДУ ДВУМЯ ПРОВОДАМИ С ОДИНАКОВЫМ ДИАМЕТРОМ, КОД 1'
30 PRINT 'МЕЖДУ ДВУМЯ ПРОВОДАМИ С РАЗНЫМИ ДИАМЕТРАМИ, КОД 2'
40 INPUT 'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ E=' E
50 INPUT 'ВВЕДИТЕ ВОЛНОВОЕ СОПРОТИВЛЕНИЕ Z=' Z
60 INPUT 'ВВЕДИТЕ ТОЧНОСТЬ РЕЗУЛЬТАТА Q=' Q
70 INPUT 'ВВЕДИТЕ КОД-' N
80 IF N=1 THEN 110
90 IF N=2 THEN 210
100 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 70
110 INPUT 'ВВЕДИТЕ ДИАМЕТР D=' D
120 INPUT 'ВВЕДИТЕ РАССТОЯНИЕ L=' L
130 LET A=0 : LET H=1
140 IF ABS(H)<Q THEN 180
150 LET A=A+H : GOSUB 190
160 IF F/H>=0 THEN 150
170 LET H=H/10 : GOTO 140
180 PRINT 'РАССТОЯНИЕ A=' A : GOTO 70
190 LET F=Z-(120/SQR(E))*((ARC(A/D)-LOG(SQR(1+(A/(2*D))^2)))
200 RETURN
210 INPUT 'ВВЕДИТЕ ДИАМЕТР D=' D
220 INPUT 'ВВЕДИТЕ ДИАМЕТР X=' X
230 LET A=0 : LET H=1
240 IF ABS(H)<Q THEN 280
250 LET A=A+H : GOSUB 290
260 IF F/H>=0 THEN 250
270 LET H=H/10 : GOTO 240
280 PRINT 'РАССТОЯНИЕ A=' A : GOTO 70
290 LET F=Z-(60/SQR(E))*((ARC((4*(A^2)-(D^2)-(X^2))/(2*D*X)))
300 RETURN : END

```

а для $D=0,1$ см, $X=0,05$ см, $Z=300$ Ом (код 2) получим $A=0,434$ см.

Линия из двух проводников одинакового диаметра D , расположенных на расстоянии A друг от друга в свободном пространстве, имеет волновое сопротивление $Z_n(Z) = \frac{120}{\sqrt{\epsilon}} \operatorname{arch} X$, где $X=A/D$. Аналогично для

линии в виде проводника с диаметром D , расположенного над проводящей плоскостью

Для $\epsilon=E=1$, $Z_{0n}=Z=200$ Ом погрешность вычислений $Q=1 \cdot 10^{-4}$, $D=0,1$ см, $L=1,5$ см (код 1) получим $A=0,2753$ см,

на расстоянии A от нее,

$$Z_n(X) = 60 \operatorname{arch} X/\sqrt{\epsilon}.$$

```

10 PRINT 'РАСЧЕТ ПАРАМЕТРА X'
20 PRINT 'ДУХПРОВОДНАЯ ЛИНИЯ, КОД 1'
30 PRINT 'ЛИНИЯ В ВИДЕ КРУГЛОГО ПРОВОДНИКА НАД ПЛОСКОСТЬЮ, КОД 2'
40 INPUT 'ВВЕДИТЕ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ E='E
50 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА Q='Q
60 INPUT 'ВВЕДИТЕ ВОЛНОВОЕ СОПРОТИВЛЕНИЕ Z='Z
70 INPUT 'ЗАДАЙТЕ КОД ЛИНИИ 'N:LETK=120:IF N=2 THEN LETK=60
80 IF N=1 THEN 110
90 IF N=2 THEN 110
100 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 70
110 LET X=0 : LET N=10
120 IF ABS(N)<Q THEN 160
130 LET X=X+N : GOSUB 170
140 IF F/N)=0 THEN 130
150 LET N=-N/10 : GOTO 120
160 PRINT 'ЗНАЧЕНИЕ X='X:GOTO 70
170 LET F=Z-(K/SQR(E))*ANC(X):RETURN:END
    
```

Для $Z_{л0}=300$ Ом, $\epsilon=E=1$ и погрешности $\epsilon=Q=1 \cdot 10^{-4}$ получим $X=6,1322$ (код 1) и $X=74,2099$ (код 2).

Большие $Z_{л}$ имеют коаксиальные линии, у которых внутренний проводник выполнен в виде спирали. Волновое сопротивление та-

При заданных $F_w=F_{w0}$, N и D параметр X находится из решения нелинейного уравнения

$$F_{w0} - F_w(X) = 0.$$

Программа 7.21.

```

10 PRINT 'РАСЧЕТ ПАРАМЕТРА X СПИРАЛЬНОЙ КОАКСИАЛЬНОЙ ЛИНИИ'
20 INPUT 'ВВЕДИТЕ N,D,E 'N,D,E:LETM=N*N
30 INPUT 'ВВЕДИТЕ Fw='A:LETX=0:LETH=.1
40 IF ABS(N)<E THEN 70
50 LETX=X+N:GOSUB 80:IF F/N)=0 THEN 50
60 LETH=-H/10:GOTO 40
70 PRINT 'X='X:GOTO 30
80 LET F=A-SQR(1+M*(#PI*X)^2*(1-(X/D)^2)/2/LOG(D/X))
90 RETURN:END
    
```

ких линий $Z_{л} = Z_{лк} F_w$, где $Z_{лк}$ — волновое сопротивление обычной коаксиальной линии со сплошным внутренним проводником, имеющим диаметр, равный внешнему диаметру спирали, а F_w — поправочный множитель

$$F_w = \sqrt{1 + N^2 \frac{(\pi X)^2}{2 \ln(D/X)} \left[1 - \left(\frac{X}{D} \right)^2 \right]}$$

где D — внутренний диаметр внешнего трубчатого проводника, X — внешний диаметр спирали, N — число витков спирали на единицу длины линии.

Для $F_{w0}=3$, $N=4$, $D=1$ см и погрешности вычислений $E=1 \cdot 10^{-3}$ получим $X=0,349$ см.

Волновое сопротивление коаксиальной линии с внутренним проводником (диаметр X), смещенным от центра на расстояние L , $Z_{л}(X) = \frac{60}{\sqrt{\epsilon}} \operatorname{arch} \frac{D^2 + X^2 - 4L^2}{2XD}$, где D — внутренний диаметр внешнего проводника. В этом случае параметр X при заданных $Z_{л}=Z_{л0}$, D , L и ϵ находится из решения нелинейного уравнения $Z_{л0} - Z_{л}(X) = 0$.

Программа 7.22.

```

10 PRINT 'РАСЧЕТ ПАРАМЕТРА X КОАКСИАЛЬНОЙ ЛИНИИ'
20 PRINT 'СО СМЕЩЕННЫМ ВНУТРЕННИМ ПРОВОДНИКОМ'
30 INPUT 'ВВЕДИТЕ ВНУТРЕННИЙ ДИАМЕТР ВНЕШНЕГО ПРОВОДНИКА D='D
40 INPUT 'ВВЕДИТЕ РАДИАЛЬНОЕ СМЕЩЕНИЕ ВНУТРЕННЕГО ПРОВОДНИКА L='L
50 INPUT 'ЗАДАЙТЕ ОТНОСИТЕЛЬНУЮ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ E='E
60 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕЗУЛЬТАТА Q='Q
70 INPUT 'ВВЕДИТЕ ВОЛНОВОЕ СОПРОТИВЛЕНИЕ ЛИНИИ Z='Z
80 LETX=0:LETH=.1
90 IF ABS(N)<Q THEN 120
100 LETX=X+N:GOSUB 130:IF F/N)=0 THEN 100
110 LETH=-H/10:GOTO 90
120 PRINT 'ДИАМЕТР ВНУТРЕННЕГО ПРОВОДНИКА X='X:GOTO 70
125 PRINT F,Z-F
130 LET F=60*ANC((D*D+X*X-4*L*L)/2/X/D)/SQR(E)
140 LET F=F-Z:RETURN:END
    
```

Для $Z_{л0} = 100 \text{ Ом}$, $D = 1 \text{ см}$, $L = 0,25 \text{ см}$, $\epsilon = 1$ и погрешности $Q = 1 \cdot 10^{-5}$ получим $X = 0,14038 \text{ см}$.

Весьма широкое распространение находят полосковые несимметричные (рис. 7.9, а) и симметричные (рис. 7.9, б) линии. Микрополосковые линии (МПЛ) изготавливаются

$$\frac{W_s}{h} = \frac{W}{h} + \frac{1,25t}{\pi h} \left(1 + \ln \frac{2h}{t}\right)$$

при $\frac{W}{h} \geq \frac{1}{2\pi}$,

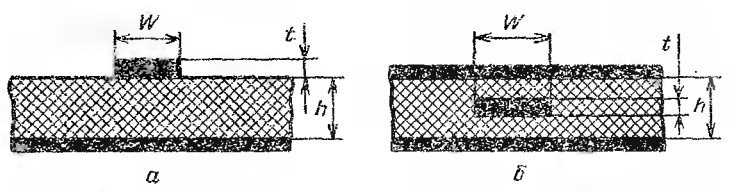


Рис. 7.9. Полосковая несимметричная (а) и симметричная (б) линии передачи (в разрезе)

в виде тонкопленочного печатного или напыленного проводника, нанесенного на диэлектрическую подложку с металлизированной нижней плоскостью (структура таких линий соответствует показанной на рис. 7.9, а). Расчет Z_n для таких линий довольно сложен, и в специальной литературе описан ряд формул, обеспечивающих расчет с различной погрешностью.

Расчет Z_n несимметричных микрополосковых линий с конечной толщиной t полоски может проводиться по формулам [2]

$$Z_n = \frac{60}{\sqrt{\epsilon_{эф}}} \ln \left(\frac{8}{W_{эф}/h} + 0,25 \frac{W_{эф}}{h} \right)$$

при $\frac{W}{h} < 1$,

$$\epsilon_{эф} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left(1 + \frac{12}{W/h}\right)^{-1/2} - C$$

при $W/h \geq 1$,

$$\epsilon_{эф} = \frac{\epsilon_r + 1}{2} + \frac{\epsilon_r - 1}{2} \left[\left(1 + \frac{12}{W/h}\right)^{-1/2} + 0,04 \left(1 - \frac{W}{h}\right)^2 \right] - C$$

при $W/h < 1$ и $C = \frac{\epsilon_r - 1}{4,6} \cdot \frac{t/h}{\sqrt{W/h}}$.

Программа 7.23.

```

10 PRINT 'РАСЧЕТ ВОЛНОВОГО СОПРОТИВЛЕНИЯ НЕСИММЕТРИЧНОЙ'
20 PRINT '    ПОЛОСКОВОЙ ЛИНИИ'
30 INPUT 'ВВЕДИТЕ ШИРИНУ ПОЛОСКИ W=' W
40 INPUT 'ВВЕДИТЕ ТОЛЩИНУ ПОЛОСКИ T=' T
50 INPUT 'ВВЕДИТЕ ТОЛЩИНУ ДИЭЛЕКТРИКА H=' H
55 INPUT 'ВВЕДИТЕ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ EPS=' E
60 LET K=W/H: LET P=H: IF K<=1/#PI/2 THEN LET P=2*#PI*W
70 LET M=K+(1,25*T/#PI/H)*(1+LOG(2*P/T))
80 LET N=0: IF K>1 THEN LET N=.04*(1-K)^2
90 LET F=(E+1)/2+(E-1)*((1+12/K)^(-.5)+N)/2
100 LET F=F-(E-1)*T/H/4.6/SQR(K)
110 IF K>=1 THEN LET 0
115 LET Z=60*LOG(8/M+M/4)/SQR(F): GOTO 140
120 LET Z=120*#PI/(M+1.393+.667*LOG(M+1.444))
130 LET Z=Z/SQR(F)
140 PRINT 'ВОЛНОВОЕ СОПРОТИВЛЕНИЕ Z0=' Z: END

```

$$Z_n = \frac{120\pi}{\sqrt{\epsilon_{эф}}} \left[\frac{W_{эф}}{h} + 1,393 + 0,667 \ln \left(\frac{W_{эф}}{h} + 1,444 \right) \right]^{-1}$$

при $W/h \geq 1$,

Контрольные примеры.

W, см	t, см	h, см	ε ₃	Z ₀ , Ом
1	0,01	1	1,17	44,653
1	0,05	1	4,3	69,223
0,1	0,01	1	1	250,426
2	0,05	1	1	87,086

где $\frac{W_s}{h} = \frac{W}{h} + \frac{1,25t}{\pi h} \left(1 + \ln \frac{4\pi W}{t}\right)$

при $\frac{W}{h} \leq \frac{1}{2\pi}$,

Ширину W и отношение фазовой скорости v к скорости света c для отмеченных выше линий при заданных Z_0 , t , h и ϵ_r

точности, достаточной для инженерных расчетов, можно вычислить с помощью формул [9]

$$W = 1,25 \left[\frac{5,98h}{\exp[Z_n(\epsilon_d + 1,41)^{1/2}/87]} - t \right],$$

$$v/c = 1 / [(0,475\epsilon_d + 0,67)^{1/2}]$$

для несимметричной линии и

$$W = 0,59 \left[\frac{4h}{\exp(Z_n \sqrt{\epsilon_d}/60)} - 2,1t \right],$$

$$v/c = 1/\sqrt{\epsilon_d}$$

— для симметричной. В этих формулах h и t задаются в миллиметрах.

Программа 7.24.

```

10 PRINT 'РАСЧЕТ W И V/C ПОЛОСКОВЫХ ЛИНИЙ'
20 PRINT 'ЛИНИЯ ПОЛОСКОВАЯ НЕСИММЕТРИЧНАЯ, КОД 1'
30 PRINT 'ЛИНИЯ ПОЛОСКОВАЯ СИММЕТРИЧНАЯ, КОД 2'
40 INPUT 'ВВЕДИТЕ КОД N'
50 INPUT 'ВВЕДИТЕ ТОЛЩИНУ ПОЛОСКИ T='T
60 INPUT 'ВВЕДИТЕ ОТНОСИТЕЛЬНУЮ ДИЭЛЕКТРИЧЕСКУЮ ПРОНИЦАЕМОСТЬ E='E
70 INPUT 'ВВЕДИТЕ ВОЛНОВОЕ СОПРОТИВЛЕНИЕ ЛИНИИ Z='Z
80 IF N=1 THEN 120
90 IF N=2 THEN 160
100 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 40
120 INPUT 'ВВЕДИТЕ ШИРИНУ ПОЛОСКИ H='H
130 LET W=1.25*((E+1.41)^(1/2)/87)-T
140 LET V=1/SQR(0.475*E+0.67)
150 GOTO 190
160 INPUT 'ВВЕДИТЕ ШИРИНУ ПОЛОСКИ B='B
170 LET A=EXP(Z*SQR(E)/60) : LET W=0.59*((4*B)/A)-2.1*T
180 LET V=1/SQR(E)
190 PRINT 'ШИРИНА ПОЛОСКИ W='W
200 PRINT 'ОТНОШЕНИЕ ФАЗОВОЙ СКОРОСТИ К СКОРОСТИ СВЕТА V/C='V
210 GOTO 40 : END

```

Для $Z_n = 75$ Ом, $t = 0,05$ мм, $h = 1$ мм и $\epsilon_d = 4,7$ получим $W = 0,825$ мм и $v/c = 0,587$ для несимметричной линии, а при $Z_n = 50$ Ом, $t = 0,5$ мм, $h = 2,5$ мм и $\epsilon_d = 4,7$ получим $W = 0,349$ мм и $v/c = 0,461$ для симметричной линии.

Для реальных длинных линий с распределенными параметрами существенны потери, связанные с конечным погонным последовательным сопротивлением R (на единицу длины) и конечной погонной шунтирующей проводимостью G . При этом элементарный участок линии длиной dx представляется эквивалентной схемой (рис. 7.10), где L и C — погонные индуктивность и емкость линии.

Для определения напряжения u и тока i в любой точке линии, лежащей на расстоянии x от ее начала, необходимо решить

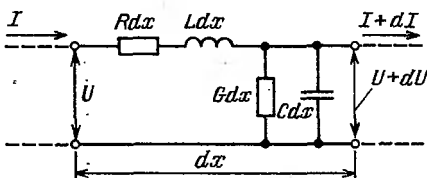


Рис. 7.10. Схема замещения элементарного участка длинной линии

систему телеграфных уравнений

$$-\frac{\partial u}{\partial x} = Ri + L \frac{\partial i}{\partial t}, \quad -\frac{\partial i}{\partial x} = Gu + C \frac{\partial u}{\partial t}.$$

Если на входе линии действует синусоидальный сигнал, то решение телеграфных уравнений имеет вид $U = Ae^{-\gamma x} + Be^{\gamma x}$, где A и B — постоянные интегрирования, а

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)} \quad (7.16)$$

— постоянная распространения. Ток в любой

точке линии $i = \frac{A}{Z_n} e^{-\gamma x} + \frac{B}{Z_n} e^{\gamma x}$, где

$$Z_n = \sqrt{\frac{Z_1}{Y_1}} = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \quad (7.17)$$

— это характеристическое сопротивление линии.

Из (7.16) следует, что $\gamma = \alpha + j\beta$, где α — постоянная затухания и β — фазовая постоянная. Постоянная α показывает, на сколько непер затухает сигнал на единицу длины (в километрах) линии, а постоянная β показывает, на сколько радian (или градусов) изменяется фаза на единицу длины линии.

Если обозначить $\varphi_1 = \arctg(\omega L/R)$ и $\varphi_2 = \arctg(\omega C/G)$, то

$$\alpha = (\sqrt{R^2 + \omega^2 L^2} \sqrt{G^2 + \omega^2 C^2})^{1/2} \cos\left(\frac{\varphi_1 + \varphi_2}{2}\right),$$

$$\beta = (\sqrt{R^2 + \omega^2 L^2} \sqrt{G^2 + \omega^2 C^2})^{1/2} \sin\left(\frac{\varphi_1 + \varphi_2}{2}\right)$$

Характеристическое сопротивление линии (7.17) можно выразить в виде

$$Z_n = |Z_n| e^{j\varphi},$$

где $|Z_n| = (\sqrt{R^2 + \omega^2 L^2} / \sqrt{G^2 + \omega^2 C^2})^{1/2}$, $\varphi = (\varphi_1 - \varphi_2)/2$.

Программа 7.25.

Для линии с $R = 167$ Ом/км, $L = 0,49$ мГн/км, $C = 0,05$ мкФ/км и $G = 1,66$ мксм/км на частоте $f = 1000$ Гц получим $|Z_n| = 729,150$ Ом, $\varphi = -44,321$ град/км, $\alpha = 0,161$ нп/км, $\beta = 0,163$ рад/км.

При использовании линий в качестве элементов задержки или накопления электри-

```

10 PRINT 'РАСЧЕТ ПОСТОЯННЫХ РАСПРОСТРАНЕНИЯ ВОЛНЫ В ДЛИННОЙ ЛИНИИ'
20 INPUT 'ВВЕДИТЕ ПОГОННОЕ СОПРОТИВЛЕНИЕ (ОМ/КМ) R='R
30 INPUT 'ВВЕДИТЕ ПОГОННУЮ ИНДУКТИВНОСТЬ (МГН/КМ) L='L
40 INPUT 'ВВЕДИТЕ ПОГОННУЮ ЕМКОСТЬ (МКФ/КМ) C='C
50 INPUT 'ВВЕДИТЕ ПОГОННУЮ ПРОВОДИМОСТЬ (МКОМ/КМ) G='G
60 LET L=L/1E3:LET G=G/1E6:LET C=C/1E6
70 INPUT 'ЗАДАЙТЕ ЧАСТОТУ (ГЦ) F='F: LET W=2*PI*F
80 LET U=SGR(R*W+(W*L)^2):LET X=SGR(G*W+(W*C)^2)
90 LET A=ATN(W*L/R):IF G=0 THEN LET B=#PI/2
100 IF G<0 THEN LET B=ATN(W*C/G)
120 PRINT 'МОДУЛЬ СОПРОТИВЛЕНИЯ /Z/'=SGR(U/X)' ОМ'
130 PRINT 'ФАЗОВЫЙ СДВИГ ='DEG((A-B)/2)' ГРАД'
140 LET Y=SGR(U*X):LET Z=(A+B)/2
150 PRINT 'ПОСТОЯННАЯ ОСЛАБЛЕНИЯ АЛЬФА='Y*W*COS(Z)
160 PRINT 'ФАЗОВАЯ ПОСТОЯННАЯ БЕТА='Y*W*SIN(Z)
170 GOTO 70:END
    
```

ческой энергии часто применяются линии с сосредоточенными постоянными, т. е. элементами L и C (рис. 7.11), в виде дискретных катушек индуктивности и конденсаторов.

Линии типа M имеют наилучшие характеристики при $M=1,27$.

Расчет искусственных линий проводится по заданному волновому сопротивлению

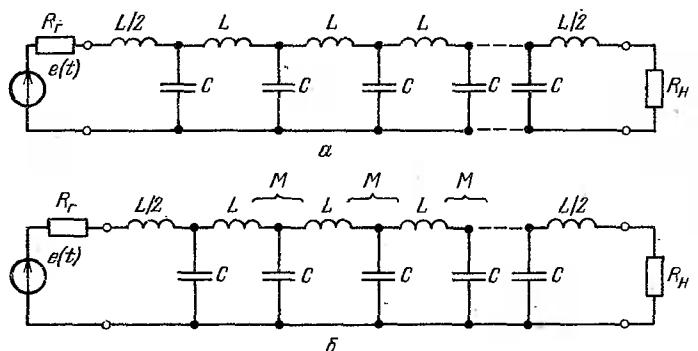


Рис. 7.11. Искусственные линии К-типа (а) и М-типа (б)

При заданном времени задержки $t_3 = n\sqrt{LC}$, где n — число звеньев линии, такие искусственные линии имеют существенно меньшие габариты и массу, чем линии с распределенными постоянными. Если между смежными индуктивностями звеньев нет магнитной связи, линии относятся к типу K , в противном случае — к типу M (M — коэффициент взаи-

$Z_n = \sqrt{L/C}$, времени нарастания импульсов на выходе линии t_0 при ее возбуждении прямоугольными импульсами и времени задержки t_3 . При этом для n , C и L можно использовать следующие выражения [9]: $n = K_1 (t_3/t_0)^{3/2}$, $C = t_3/K_2 n Z_n$ и $L = t_3 Z_n / K_2 n$.

Программа 7.26.

```

10 PRINT 'РАСЧЕТ ПАРАМЕТРОВ ИСКУССТВЕННЫХ ЛИНИЙ ЗАДЕРЖКИ'
20 PRINT 'ЛИНИЯ НА ФИЛЬТРАХ К-ТИПА, КОД 1'
30 PRINT 'ЛИНИЯ НА ФИЛЬТРАХ М-ТИПА, КОД 2'
40 INPUT 'ВВЕДИТЕ КОД' N
50 IF N=1 THEN 60
55 GOTO 70
60 LET K1=1.1:LET K2=1.07:GOTO 100
70 IF N<2 THEN 90
80 LET K1=0.94:LET K2=1.2:GOTO 100
90 PRINT 'КОД НАБРАН НЕВЕРНО':GOTO 40
100 INPUT 'ВВЕДИТЕ ВРЕМЯ ЗАДЕРЖКИ T2='T1
110 INPUT 'ВВЕДИТЕ ДЛИТЕЛЬНОСТЬ ФРОНТА TF='T2
120 INPUT 'ВВЕДИТЕ ВОЛНОВОЕ СОПРОТИВЛЕНИЕ Z='Z
130 LET N=K1*((T1/T2)^1.5)
135 PRINT 'ЧИСЛО ЗВЕНЬЕВ N='N
140 INPUT 'ВВЕДИТЕ УТОЧНЕННОЕ ЗНАЧЕНИЕ N='N
150 LET L=(T1*Z)/(N*K2)
160 LET C=T1/(K2*N*Z)
170 PRINT 'ИНДУКТИВНОСТЬ ЗВЕНЬЕВ L='L
180 PRINT 'ЕМКОСТЬ ЗВЕНЬЕВ C='C
190 GOTO 40:END
    
```


Пример. Для $Z_n=600$ Ом, $t_3=1 \cdot 10^{-6}$ с и $t_\phi=5 \cdot 10^{-8}$ с получим $n=84,076$. Округлив n до 85, находим (код 2) $L=5,882 \cdot 10^{-6}$ Гн и $C=1,634 \cdot 10^{-11}$ Ф. Если при тех же исходных данных указать код 1 (линия типа К), то получим $n=98,384$. Округлив n до 100, находим $L=5,607 \cdot 10^{-6}$ Гн и $C=1,558 \cdot 10^{-11}$ Ф.

$=R=23,333$ Ом, $r_6=RB=83,333$ Ом, $r_k=$
 $=RK=3,3325 \cdot 10^5$ Ом и $\alpha=A=0,97999$.
 Для схемы с общим эмиттером (код 2) при $h_{11}=2800$ Ом, $h_{12}=2 \cdot 10^{-3}$ См, $h_{21}=60$ и $h_{22}=5 \cdot 10^{-5}$ См получим $r_3=R=40$ Ом, $r_6=RB=360$ Ом, $r_m=RM=1,22 \cdot 10^6$ Ом и $\beta=A=0,9836$.

Программа 7.28.

Пример. Для схемы с общим эмиттером (код 1) при $r_3=R1=25$ Ом, $r_6=R2=400$ Ом, $r_k=R3=1 \cdot 10^6$ Ом и $\alpha=A=0,985$ получим $h_{11}=2,064 \cdot 10^3$ Ом, $h_{12}=1,664 \cdot 10^{-3}$, $h_{21}=$
 $=65,556$ и $h_{22}=6,656 \cdot 10^{-5}$ См. Для схемы с общей базой (код 2) и указанных выше исходных данных получим $h_{11}=30,998$ Ом,

§ 7.5. Расчет усилителей

При расчете усилителей на биполярных транзисторах бывает необходимо вычислять физические параметры транзистора по h -параметрам (табл. 7.2) и наоборот (табл. 7.3) [34].

Таблица 7.2

Формулы для расчета физических параметров биполярного транзистора по h -параметрам соответствующей схемы включения

Физический параметр	Схема с общей базой	Схема с общим эмиттером
Сопротивление эмиттера r_3	$h_{11} = \frac{(1+h_{21})h_{12}}{h_{22}}$	h_{12}/h_{22}
Сопротивление базы r_6	h_{12}/h_{22}	$h_{11} = \frac{h_{12}(1+h_{21})}{h_{22}}$
Сопротивление коллектора r_k	$\frac{1-h_{12}}{h_{22}}$	$\frac{1+h_{21}}{h_{22}}$
Сопротивление $r_m = ar_k$	$\frac{h_{12}+h_{21}}{h_{22}}$	$\frac{h_{12}+h_{21}}{h_{22}}$
Коэффициент передачи тока эмиттера a	$\frac{h_{12}+h_{21}}{1-h_{12}}$	$\frac{h_{12}+h_{21}}{1+h_{21}}$

Таблица 7.3

Формулы для расчета h -параметров биполярного транзистора по физическим параметрам

h -параметр	Схема включения		
	с общим эмиттером	с общей базой	с общим коллектором
h_{11}	$r_6 + \frac{r_3 r_k}{r_3 + r_k - ar_k}$	$r_3 + \frac{(1-a)r_6 r_k}{r_6 + r_k}$	$r_6 + \frac{r_3 r_k}{r_3 + r_k - ar_k}$
h_{12}	$\frac{r_3}{r_3 + r_k - ar_k}$	$\frac{r_6}{r_k + r_6}$	$\frac{r_k - ar_k}{r_3 + r_k - ar_k}$
h_{21}	$\frac{ar_k - r_3}{r_3 + r_k - ar_k}$	$\frac{r_6 + ar_k}{r_6 + r_k}$	$\frac{r_k}{r_3 + r_k - ar_k}$
h_{22}	$\frac{1}{r_3 + r_k - ar_k}$	$\frac{1}{r_k + r_6}$	$\frac{1}{r_3 + r_k - ar_k}$

Программа 7.27.

Пример. Для схемы с общей базой (код 1) при $h_{11}=25$ Ом, $h_{12}=0,25 \cdot 10^{-3}$ См, $h_{21}=-0,98$ и $h_{22}=3 \cdot 10^{-6}$ См получим $r_3=$

$h_{12}=3,998 \cdot 10^{-4}$, $h_{21}=-0,985$ и $h_{22}=$
 $=9,996 \cdot 10^{-7}$ См. Наконец, для схемы с общим коллектором (код 3) получим $h_{11}=$
 $=2,064 \cdot 10^3$ Ом, $h_{12}=0,9983$, $h_{21}=-66,556$ и $h_{22}=6,656 \cdot 10^{-5}$ См.

Программа 7.27.

```

10 PRINT 'РАСЧЕТ ФИЗИЧЕСКИХ ПАРАМЕТРОВ ТРАНЗИСТОРА ПО Н-ПАРАМЕТРАМ'
20 PRINT 'СХЕМА С ОБЩЕЙ БАЗОЙ, КОД 1'
30 PRINT 'СХЕМА С ОБЩЕЙ ЭМИТТЕРОМ, КОД 2'
40 INPUT 'ВВЕДИТЕ ВХОДНОЕ СОПРОТИВЛЕНИЕ H11='H1
50 INPUT 'ВВЕДИТЕ КОЭФФИЦИЕНТ ОБРАТНОЙ СВЯЗИ ПО НАПРЯЖЕНИЮ H12='H2
60 INPUT 'ВВЕДИТЕ КОЭФФИЦИЕНТ ПЕРЕДАЧИ ПО ТОКУ H21='H3
70 INPUT 'ВВЕДИТЕ ВЫХОДНУЮ ПРОВОДИМОСТЬ H22='H4
80 INPUT 'ВВЕДИТЕ КОД' N
90 IF N=1 THEN 120
100 IF N=2 THEN 180
110 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 80
120 LETR1=H1-(1+H3)*H2/H4:LETR2=H2/H4:LETR3=(1-H2)/H4
150 LETR4=(H2+H3)/H4:LETA=(H2+H3)/(1-H2):GOTO 230
180 LETR1=H2/H4:LETR2=H1-H2*(1+H3)/H4:LETR3=(1+H3)/H4
200 LETR4=(H2+H3)/H4:LETA=(H2+H3)/(1+H3)
230 PRINT 'СОПРОТИВЛЕНИЕ ЭМИТТЕРА R='R1
240 PRINT 'СОПРОТИВЛЕНИЕ БАЗЫ RB='R2
250 PRINT 'СОПРОТИВЛЕНИЕ КОЛЛЕКТОРА RK='R3
260 PRINT 'СОПРОТИВЛЕНИЕ RM='R4
270 PRINT 'КОЭФФИЦИЕНТ ПЕРЕДАЧИ ТОКА ЭМИТТЕРА A='A
280 GOTO 40:END
    
```

Программа 7.28.

```

10 PRINT 'РАСЧЕТ Н-ПАРАМЕТРОВ ПО ФИЗИЧЕСКИМ ПАРАМЕТРАМ ТРАНЗИСТОРА'
20 PRINT 'СХЕМА С ОБЩИМ ЭМИТТЕРОМ, КОД 1'
30 PRINT 'СХЕМА С ОБЩЕЙ БАЗОЙ, КОД 2'
40 PRINT 'СХЕМА С ОБЩИМ КОЛЛЕКТОРОМ, КОД 3'
50 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ ЭМИТТЕРА R='R1
60 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ БАЗЫ RB='R2
70 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ КОЛЛЕКТОРА RK='R3
80 INPUT 'ВВЕДИТЕ КОЭФФИЦИЕНТ ПЕРЕДАЧИ ТОКА ЭМИТТЕРА A='A
90 INPUT 'ВВЕДИТЕ КОД' N
100 IF N=1 THEN 140
110 IF N=2 THEN 190
120 IF N=3 THEN 240
130 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 90
140 LET Y=R1+R3-A*R3 : LET H1=R2+R1*R3/Y
150 LET H2=R1/Y:LETH3=(A*R3-R1)/Y:LETH4=1/Y:GOTO 280
190 LET M=R2+R3 : LET H1=R1+((1-A)*R2*R3)/M
200 LET H2=R2/M:LETH3=-R2/A/R3:LETH4=1/M:GOTO 280
240 LET Y=R1+R3-A*R3 : LET H1=R2+R1*R3/Y
250 LET H2=R3*(1-A)/Y:LETH3=-R3/Y:LETH4=1/Y
280 PRINT 'ВХОДНОЕ СОПРОТИВЛЕНИЕ H11='H1
290 PRINT 'КОЭФФИЦИЕНТ ОБРАТНОЙ СВЯЗИ ПО НАПРЯЖЕНИЮ H12='H2
300 PRINT 'КОЭФФИЦИЕНТ ПЕРЕДАЧИ ПО ТОКУ H21='H3
310 PRINT 'ВЫХОДНАЯ ПРОВОДИМОСТЬ H22='H4
320 GOTO 90:END
    
```

Параметрами элементарных усилительных каскадов (см. рис. 7.12 и эквивалентные схемы рис. 7.13) на средних частотах являются: входное сопротивление $R_{вх} = U_{вх}/I_{вх}$, коэффициент усиления по току $K_I = I_{вых}/I_{вх}$,

коэффициент усиления по напряжению $K_U = U_{вых}/U_{вх}$, сквозной коэффициент усиления по напряжению: $K_E = U_{вых}/E_{вх}$ и выходное сопротивление $R_{вых} = \Delta U_{вых}/\Delta I_{вых}$, где $\Delta I_{вых}$ — изменение выходного тока, вызванное изме-

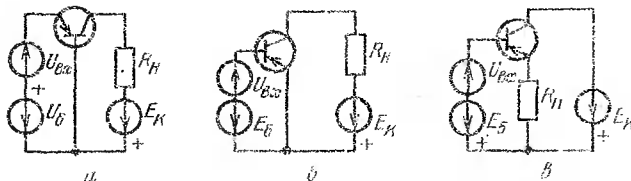


Рис. 7.12. Элементарные каскады на биполярных транзисторах с общей базой (а), общим эмиттером (б) и общим коллектором (в)

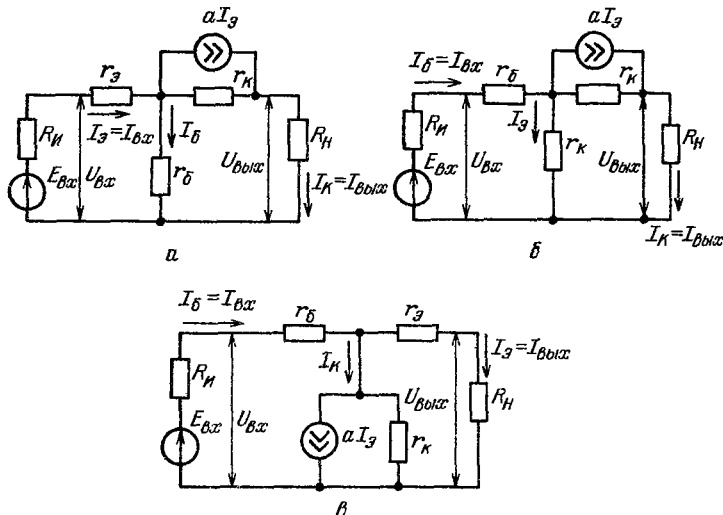


Рис. 7.13. Эквивалентные схемы каскадов с общей базой (а), общим эмиттером (б) и общим коллектором (в)

нием нагрузки. Резистивные цепи питания каскадов на рис. 7.12 и 7.13 не показаны, их учет на параметры каскада тривиален. Приведем формулы для расчета указанных параметров для каскада с общей базой:

$$R_{вх} = r_3 + r_6 \frac{(1-a)r_k + R_n}{r_k + r_6 + R_n}$$

$$K_I = \frac{ar_k + r_6}{r_k + r_6 + R_n}$$

$$K_U = \frac{(ar_k + r_6) R_n}{r_3(r_k + r_6 + R_n) + r_6[(1-a)r_k + R_n]}$$

$$K_E = \frac{(ar_k + r_6) R_n}{(R_n + r_3)(r_k + r_6 + R_n) + r_6[(1-a)r_k + R_n]}$$

$$R_{вых} = r_k + r_6 \left(1 - \frac{ar_k + r_6}{r_3 + r_6 + R_n}\right)$$

для каскада с общим эмиттером:

$$R_{вх} = r_6 + \frac{r_3(r_k + R_n)}{r_k(1-a) + r_3 + R_n}$$

$$K_I = \frac{-(ar_k - r_3)}{(1-a)r_k + r_3 + R_n}$$

$$K_U = \frac{-(ar_k - r_3) R_n}{r_3(r_k + R_n) + r_6[(1-a)r_k + r_3 + R_n]}$$

$$K_E = \frac{-(ar_k - r_3) R_n}{r_3(r_k + R_n) + (R_n + r_6)[(1-a)r_k + r_3 + R_n]}$$

$$R_{вых} = r_k(1-a) + r_3 \left(1 + \frac{ar_k + r_3}{r_3 + r_6 + R_n}\right)$$

для каскада с общим коллектором:

$$R_{вх} = r_6 + \frac{r_k(r_3 + R_n)}{r_k(1-a) + r_3 + R_n}$$

$$K_I = \frac{r_k}{r_k(1-a) + r_3 + R_n}$$

$$K_U = \frac{r_k R_n}{r_k(r_3 + R_n) + r_6[r_k(1-a) + r_3 + R_n]}$$

$$K_E = \frac{r_k R_n}{r_k(r_3 + R_n) + (r_6 + R_n)[r_k(1-a) + r_3 + R_n]}$$

$$R_{вых} = r_3 + \frac{(r_6 + R_n)r_k(1-a)}{r_k + r_6 + R_n}$$

Программа 7.29.

Пример. Для схемы с общей базой (код 1) при $r_6=400$ Ом, $r_3=25$ Ом, $r_k=1 \cdot 10^6$ Ом, $R_n=34,946$ Ом, $a=2,534 \cdot 10^5$ Ом получим $K_I=0,9751$, $K_U=139,519$ и $K_E=36,130$. Для схемы с общим эмиттером (код 2) при $r_6=400$ Ом, $r_3=25$ Ом, $r_k=1 \cdot 10^6$ Ом, $R_n=1000$ Ом, $a=0,98$ и $R_n=5000$ Ом расчет дает $R_{вх}=1403,996$ Ом, $R_{вых}=3,722 \cdot 10^4$ Ом, $K_I=-39,16$, $K_U=-139,459$ и $K_E=-81,447$ (знак — указывает на инвертирование фазы выходного сигнала по отношению к фазе входного сигнала). Для схемы с общим коллектором (код 3) при $r_6=400$ Ом, $r_3=25$ Ом, $r_k=1 \cdot 10^6$ Ом, $R_n=1 \cdot 10^4$ Ом, $a=0,98$ и $R_n=500$ Ом получаем $R_{вх}=2,598 \cdot 10^4$ Ом, $R_{вых}=2,309 \cdot 10^2$ Ом, $K_I=48,721$, $K_U=0,9377$ и $K_E=0,6771$.

Расчет по приведенной методике дает параметры каскадов в области средних частот. Искажения в области низких и высоких частот при активной нагрузке легко учитываются (см. [34]). Если нагрузка комплексная, расчет несколько усложняется. В качестве примера рассмотрим инженерный расчет параметров каскадов с общим истоком и стоком на полевых транзисторах [40].

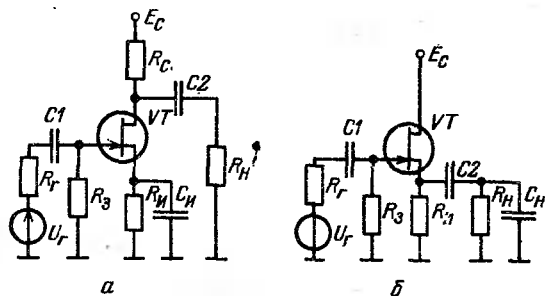
Каскад с общим истоком (рис. 7.14, а) рассчитывается по следующему алгоритму.

1. Задаем исходные данные: сопротивление затвора R_z , сопротивление стока R_c , выходное сопротивление транзистора $R_{сн}$, сопротивление генератора R_g , сопротивление нагрузки

```

10 PRINT 'РАСЧЕТ МАЛОСИГНАЛЬНЫХ УСИЛИТЕЛЬНЫХ КАСКАДОВ'
20 PRINT 'КАСКАД С ОБЩЕЙ БАЗОЙ, КОД 1'
30 PRINT 'КАСКАД С ОБЩИМ ЭМИТТЕРОМ, КОД 2'
40 PRINT 'КАСКАД С ОБЩИМ КОЛЛЕКТОРОМ, КОД 3'
50 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ БАЗЫ R1='R1
60 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ ЭМИТТЕРА R2='R2
70 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ КОЛЛЕКТОРА R3='R3
75 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ ИСТОЧНИКА R='R
80 INPUT 'ВВЕДИТЕ КОЭФФИЦИЕНТ ПЕРЕДАЧИ ТОКА ЭМИТТЕРА A='A
85 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ НАГРУЗКИ RN='R0
90 INPUT 'ВВЕДИТЕ КОД' N
100 IF N=1 THEN 140
110 IF N=2 THEN 200
120 IF N=3 THEN 260
130 PRINT 'КОД НАБРАН НЕВЕРНО' : GOTO 90
140 LET Y=R3+R1+R0 : LET R4=R2+R1*((1-A)*R3+R0)/Y
150 LET X=A*R3+R1 : LET K1=X/Y
160 LET K2=(X*R0)/(R2*Y+R1*R3*(1-A)+R0*R1)
170 LET K3=X*R0/((R+R2)*Y+R1*R3*(1-A)+R0*R1)
180 LET R5=R3+R1*(1-X/(R2+R1+R))
190 GOTO 310
200 LET M=R3*(1-A)+R2+R0 : LET R4=R1+R2*(R3+R0)/M
210 LET N=A*R3-R2 : LET K1=-N/M
220 LET K2=-N*R0/(R2*(R3+R0)+R1*M)
230 LET K3=-N*R0/((R2*(R3+R0)+(R1+R)*M))
240 LET R5=R3*(1-A)+R2*(1+(A*R3+R2)/(R2+R1+R))
250 GOTO 310
260 LET X=R2+R0 : LET Y=R3*(1-A)+X : LET R4=R1+R3*X/Y
270 LET K1=R3/Y
280 LET K2=R3*R0/(R3*X+R1*Y)
290 LET K3=R3*R0/(R3*X+(R1+R)*Y)
300 LET R5=R2+(R1+R)*(R3*(1-A)/(R3+R1+R))
310 PRINT 'ВХОДНОЕ СОПРОТИВЛЕНИЕ R='R4
320 PRINT 'ВЫХОДНОЕ СОПРОТИВЛЕНИЕ R='R5
330 PRINT 'КОЭФФИЦИЕНТ УСИЛЕНИЯ ПО ТОКУ K1='K1
340 PRINT 'КОЭФФИЦИЕНТ УСИЛЕНИЯ ПО НАПРЯЖЕНИЮ KU='K2
350 PRINT 'СКВОЗНОЙ КОЭФФИЦИЕНТ УСИЛЕНИЯ ПО НАПРЯЖЕНИЮ KZ='K3
360 GOTO 90 : END
    
```

Рис. 7.14. Каскады с общим истоком (а) и общим стоком (б) на полевых транзисторах



R_n , сопротивление в цепи истока R_n , входную емкость $C_{зп}$, проходную емкость $C_{зс}$, выходную емкость $C_{сн}$, емкости разделительных конденсаторов C_1 и C_2 , емкость шунтирующего R_n конденсатора C_n и крутизну транзистора S в рабочей точке.

2. Вычисляем входное $R_{вх} = R_3$ и выходное $R_{вых} = R_{сн} \parallel R_c = R_{сн} R_c / (R_{сн} + R_c)$ сопротивления каскада.

3. Вычисляем коэффициенты усиления

$$K_{U_n} = \frac{U_n}{U_z} = S \frac{R_{сн} R_c R_n}{R_{сн} R_c + R_{сн} R_n + R_c R_n},$$

$$K_U = \frac{U_n}{U_r} = K_{U_0} \frac{R_3}{R_3 + R_r}.$$

4. Находим характеристические постоянные времени для области малых времен:

$$\begin{aligned}
 \tau_{вх} &= |C_{зп} + C_{зс}(1 + K_{U_0})| R_r R_3 / (R_r + R_3), \\
 \tau_{вых} &= (R_c \parallel R_n \parallel R_{сн}) (C_{зс} + C_{сн}), \\
 \tau_{вх.0} &= R_r [C_{зп} + C_{зс} C_{сн} / (C_{зс} + C_{сн})], \\
 \tau_{эв} &= \tau_{вх} + \tau_{вых}.
 \end{aligned}$$

5. Вычисляем коэффициент $m = \tau_{вх.0} / \tau_{эв}^2$, характеризующий режим работы каскада.

Для данного каскада при больших C_n (C_n выбирается исходя из подавления отрицательной связи на низких частотах) значения $m < 0,25$ и режим каскада аperiодический.

6. Вычисляем времена нарастания каскада $t_n = 2,2\tau_{эвк}\sqrt{1-2m}$ и добротность $D = S/\sqrt{2}(C_{эи} + C_{сн})$.

7. Переходная характеристика каскада в области малых времен в операторной форме имеет вид $h(p) = (1 + p\tau_{эвк} + p^2\tau_{эвк}^2 m)^{-1}$.

8. Находим постоянные времена каскада в области больших времен $\tau_1 = (R_r + R_s) C_1$, $\tau_2 = (R_c + R_n) C_2$, $\tau_n = (R_n \| 1/S) C_n$.

9. По заданной длительности импульсов t_n оцениваем спады вершины импульсов $\Delta_1 = t_n/\tau_1$, $\Delta_2 = t_n/\tau_2$, $\Delta_3 = t_n/\tau_n$ и общий спад $\Delta = \Delta_1 + \Delta_2 + \Delta_3$. Если задано $R_n = 0$, то следует полагать $\Delta_3 = 0$.

Программа 7.30.

```

10 PRINT 'УСИЛИТЕЛЬ С ОБЩИМ ИСТОКОМ'
20 INPUT 'ВВЕДИТЕ R3, RC, RCH, R1, R2, R3'
30 INPUT 'ВВЕДИТЕ RF, RH, RH', R4, R5, R8
40 INPUT 'ВВЕДИТЕ C3H, C3C, CCH', C1, C2, C3
45 INPUT 'ВВЕДИТЕ C1, C2, CH', C4, C5, C6
50 INPUT 'ВВЕДИТЕ S=' S
60 LET R6=R1:LET R7=R3*R2/(R3+R2)
70 PRINT 'R ВХОДА=R6:PRINT 'R ВХОДА=' R7
80 LET A=R2*R3*R5/(R2*R3+R2*R5+R3*R5)
90 LET K=(R1/(R1+R4))*S*A:PRINT 'KU=' K
100 LET K1=S*A:LET T1=(C1+C2*(1+K1))*R4*R1/(R4+R1)
110 LET T2=(C2+C3)*A:LET T3=R4*(C1+C2*C3/(C2+C3))
115 PRINT 'T1=' T1, 'T2=' T2
120 LET T4=T1+T2:LET M=T3*T2/T4^2
130 PRINT 'H(P)=1/(1+P*T4+P^2*T4^2*M), ГДЕ'
140 PRINT 'T4=' T4:PRINT 'M=' M
150 LET T5=2.2*T4*SQR(1-2*M):LET D=S/2.2/(C1+C3)
160 PRINT 'T НАРАСТАНИЯ=' T5:PRINT 'ДОБРОТНОСТЬ D=' D
170 PRINT 'ДЛЯ ОБЛАСТИ БОЛЬШИХ ВРЕМЕН'
180 LET T6=(R4+R1)*C4:LET T7=(R2+R5)*C5
190 LET T8=C6*R8/(R8*S+1)
200 PRINT 'T6=' T6, 'T7=' T7, 'T ИСТ. ЦЕПИ=' T8
210 INPUT 'ВВЕДИТЕ T ИМПУЛЬСА=' T0
220 LET Z1=T0/T6:LET Z2=T0/T7:LET Z3=0:IF T8>0 THEN LET Z3=T0/T8
230 LET Z=Z1+Z2+Z3
240 PRINT 'СПАД ПЛОСКОК ВЕРШИНЫ ИМПУЛЬСА Z=' Z
250 GOTO 210:END

```

Пример. Для $R_3 = 1 \cdot 10^6$ Ом, $R_c = 2 \cdot 10^3$ Ом, $R_{сн} = 1 \cdot 10^6$ Ом, $R_r = 1 \cdot 10^8$ Ом, $R_n = 1 \cdot 10^4$ Ом, $R_n = 1 \cdot 10^3$ Ом, $C_{эи} = 10 \cdot 10^{-12}$ Ф, $C_{сн} = 1 \cdot 10^{-12}$ Ф, $C_{сн} = 5 \cdot 10^{-12}$ Ф, $C_1 = 0,1 \cdot 10^{-6}$ Ф, $C_2 = 1 \cdot 10^{-6}$ Ф, $C_n = 100 \cdot 10^{-6}$ Ф, $S = 5 \cdot 10^{-3}$ А/В получим $R_{вх} = 1 \cdot 10^6$ Ом, $R_{вх} = 1,961 \cdot 10^3$ Ом, $K_U = 8,189$, $\tau_{вх} = 1,918 \cdot 10^{-8}$ с, $\tau_{вх} = 9,836 \cdot 10^{-9}$ с, $\tau_{эвк} = T4 = 2,901 \cdot 10^{-8}$, $m = M = 1,266 \cdot 10^{-1}$, $t_n = 5,516 \cdot 10^{-8}$ с, $D = 1,515 \cdot 10^5$ 1/с, $\tau_1 = 1,001 \cdot 10^{-1}$ с, $\tau_2 = 1,2 \cdot 10^{-2}$ с, $\tau_n = 1,667 \cdot 10^{-2}$ с, для $t_n = 100 \cdot 10^{-6}$ с, $\Delta = Z = 1,533 \cdot 10^{-2}$.

Каскад с общим стоком (истоковый повторитель), работающий на емкостную нагрузку (рис. 7.14, б), рассчитывается по следующему алгоритму.

1. Задаем исходные данные: сопротивление затвора R_3 , истока R_n , генератора R_r , нагрузки R_n , емкости $C_{эи}$, $C_{сн}$ и C_n (емкость C_n включает и емкость $C_{сн}$) и крутизну S .

2. Находим коэффициенты передачи по напряжению $K_{U0} = SR_n/(1 + SR_n)$, где $R_{ин} = R_n R_n / (R_n + R_n)$, и $K_U = R_s K_{U0} / (R_s + R_r)$, входное $R_{вх} = R_s$ и выходное $R_{вх} = R_n \| (1/S) = R_n / (1 + SR_n)$ сопротивления.

3. Находим характеристики постоянные времени в области малых времен $\tau_{вх} = R_r [C_{эи} (1 - K_U) + C_{сн}]$, $\tau_{вх} = (R_n \| R_n) \| (1/S) C_n$, $\tau_{вх0} = R_r (C_{эи} + C_{сн})$ и $\tau_{эвк} = \tau_{вх} + \tau_{вх0}$.

4. Определяем коэффициент режима $m = \tau_{вх0} \tau_{вх} / \tau_{эвк}^2$.

5. Если $m \leq 0,25$, режим работы является апериодическим (при $m = 0,25$ — критическим), причем время нарастания каскада $t_n = 2,2\tau_{эвк}\sqrt{1-2m}$.

6. Если $m > 0,25$, режим работы является колебательным и операторное выражение для переходной характеристики имеет вид $h(p) = (1 + p\tau_{эвк} + p^2\tau_{эвк}^2 m)^{-1}$, где значения $\tau_{эвк}$ и m соответствуют полученным выше.

При необходимости могут вычисляться постоянные времена в области больших времен $\tau_1 = C_1 (R_r + R_s)$, $\tau_2 = C_2 (R_{вх} + R_n)$ и относительный спад вершины импульса $\Delta = t_n/\tau_1 + t_n/\tau_2$ заданной длительности t_n .

Программа 7.31.

Пример. Для $R_3 = 1 \cdot 10^6$ Ом, $R_n = 2 \cdot 10^3$ Ом, $R_r = 1 \cdot 10^8$ Ом, $R_n = 1 \cdot 10^4$ Ом, $C_{эи} = 10 \cdot 10^{-12}$ Ф, $C_{сн} = 1 \cdot 10^{-12}$ Ф, $C_n = 100 \cdot 10^{-12}$ Ф и $S = 5 \cdot 10^{-3}$ А/В получим $R_{вх} = 1 \cdot 10^6$ Ом, $R_{вх} = 1,818 \cdot 10^2$ Ом, $K_U = 0,892$, $\tau_{вх} = 2,071 \cdot 10^{-9}$ с, $\tau_{вх} = 1,790 \cdot 10^{-8}$ с, $\tau_{эвк} = 1,993 \cdot 10^{-8}$ с, режим работы колебательный при $m = 0,495$ (программа выдает формулу для $h(p)$ при $\tau_{эвк} = T4$).

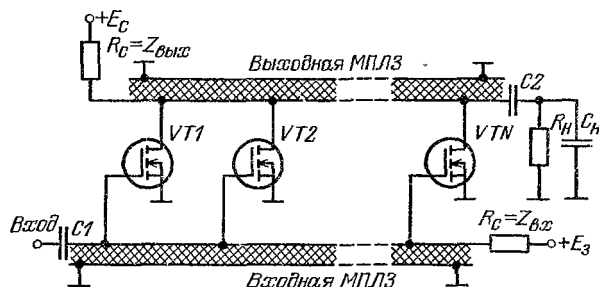
Для усиления импульсных сигналов с малыми временами нарастания и спада (порядка 1 нс) применяются усилители с распределенным усилением. Расчет такого усилителя на мощных СВЧ МДП-транзисторах с микрополосковыми линиями задержки (рис. 7.15) производится по алгоритму.

1. Задаем исходные данные: входное и выходное сопротивления линий $Z_{вх}$ и $Z_{вх}$, число транзисторов в каскаде N , их кру-

```

10 PRINT'РАСЧЕТ ИСТОКОВОГО ПОВТОРИТЕЛЯ'
20 INPUT'ВВЕДИТЕ R3, R4, R5, R6, R1, R2, R3, R4
30 INPUT'ВВЕДИТЕ C3H, C3C, CH, C1, C2, C3
40 INPUT'ВВЕДИТЕ S='S
50 LETR5=R1:LETR6=R2/(R2*S+1)
60 PRINT'R ВЫХОДА='R5:PRINT'R ВЫХОДА='R6
70 LET A=S*R2*R4/(R2+R4+S*R2*R4)
80 LET K=R1*A/(R1+R3):PRINT'КУ='K
90 LET K1=A:LET T1=R3*(C1*(1-K1)+C2)
100 LETT2=C3*R6*R4/(R6+R4):LETT3=R3*(C1+C2)
110 LET T4=T1+T2:LET M=T3*T2/T4^2
115 PRINT'TBX='T1:PRINT'TBX='T2:PRINT'TXB='T4
120 IF M>.25 THEN 180
130 IF M<.25 THEN 150
140 PRINT'РЕЖИМ РАБОТЫ КРИТИЧЕСКИЙ':GOTO 160
150 PRINT'РЕЖИМ РАБОТЫ АПЕРИОДИЧЕСКИЙ'
160 LET T5=2.2*T4*SQR(1-2*M)
170 PRINT'T НАРАСТАНИЯ='T5:GOTO 190
180 PRINT'РЕЖИМ РАБОТЫ КОЛЕБАТЕЛЬНЫЙ'
190 PRINT'K(P)=1/(1+P*T4+P^2*T4^2*M), ГДЕ'
200 PRINT'T4='T4:PRINT'M='M:END
    
```

Рис. 7.15. Усилитель с распределенным усилением на мощных МДП-транзисторах с микрополосковыми линиями задержки



тизну S , емкость нагрузки C_n , выходную $C_{\text{вых}} = C_{\text{си}}$ и проходную $C_{\text{проход}} = C_{\text{зс}}$ емкости транзисторов, диэлектрическую проницаемость $\epsilon = E$ и толщину H диэлектрика печатной платы, толщину металлических полосок микрополосковых линий D .

2. Вычисляем коэффициенты усиления $K_{\text{ус}} = NSZ_{\text{вых}}Z_n / (Z_{\text{вых}} + Z_n)$, $K_{\text{ус макс}} = NSZ_{\text{вых}}$.
3. Находим время установления в режиме согласования ($Z_n = Z_{\text{вых}}$), где $C_0 = C_{\text{зи}} + C_{\text{зс}}$

$$t_y = 1,1 Z_{\text{вых}} (C_{\text{зи}} + C_{\text{зс}}) \sqrt{N} \quad \text{при } C_n = 0,$$

$$t_y = 2,2 Z_{\text{вых}} \sqrt{N C_0^2 + C_n^2} \quad \text{при } C_n > 0.$$

4. Находим ширину полоски линий $W =$

$$= 1,25 \left[\frac{5,9H}{\exp(Z\sqrt{\epsilon + 1,41/87})} \right] - D, \text{ где } Z = Z_{\text{вх}}$$

для входной и $Z = Z_{\text{вых}}$ для выходной линий.

5. Находим относительную скорость распространения сигнала в линиях $v/c =$

$$= 1/\sqrt{0,475\epsilon + 0,67}, \text{ где } c \text{ — скорость света.}$$

Расстояние l между транзисторами должно быть таким, чтобы время прохождения сигнала между ними было равно или превышало значение t_y при $C_n = 0$.

Программа 7.32.

```

10 PRINT'УРУ НА МОШНЫХ МДП-ТРАНЗИСТОРАХ И МЛЗ'
20 INPUT'ВВЕДИТЕ ВОЛНОВЫЕ СОПРОТИВЛЕНИЯ ZВХ, ZВЫХ' Z1, Z2
30 INPUT'ВВЕДИТЕ ЧИСЛО ТРАНЗИСТОРОВ В КАСКАДЕ N'N
40 INPUT'ВВЕДИТЕ КРУТИЗНУ ТРАНЗИСТОРА S'S
50 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ СХЕМЫ RH, CH, R1, C1
60 INPUT'ВВЕДИТЕ СВЫХ, СПРОХОДА ТРАНЗИСТОРА C2, C3
70 INPUT'ВВЕДИТЕ ДИЭЛ. ПРОНИЦАЕМОСТЬ E'E
80 INPUT'ВВЕДИТЕ ТОЛЩИНУ ДИЭЛЕКТРИКА H'H
90 INPUT'ВВЕДИТЕ ТОЛЩИНУ МЕТАЛ. ПОЛОСКИ D'D:PRINT
100 LET K=N*S*Z2*R1/(Z2+R1):LET K1=N*S*Z2
110 PRINT'КУС='K:PRINT'КУС МАКС='K1
120 LET C=C2+C3:IF C1=0 THEN 140
130 LET T1=2.2*Z2*SQR(N*C^2+C1^2):GOTO 150
140 LET T1=1.1*Z2*C*SQR(N)
150 PRINT'TУСТ='T1:LET A1=EXP(Z1*SQR(E+1.41)/87)
160 LET A2=EXP(Z2*SQR(E+1.41)/87)
170 LET W1=1.25*(5.98*H/A1-D):LET W2=1.25*(5.98*H/A2-D)
180 PRINT'ШИРИНА ВХ. ЛИНИИ='W1:'ШИРИНА ВЫХ. ЛИНИИ='W2
190 LET V=1/SQR(.475*E+.67)
200 PRINT'ОТНОСИТ. СКОРОСТЬ РАСПРОСТРАНЕНИЯ='V:END
    
```

Пример. Для $Z_{вх} = 50 \text{ Ом}$, $Z_{вых} = 75 \text{ Ом}$, $N = 5$, $S = 0,025 \text{ А/В}$ (транзисторы КП905), $R_n = 75 \text{ Ом}$, $C_n = 0$, $C_{вх} = 1,6 \cdot 10^{-12} \text{ Ф}$, $C_{проход} = 0,4 \cdot 10^{-12} \text{ Ф}$, $\varepsilon = 4,7$, $H = 1 \text{ мм}$ и $d = 0,1 \text{ мм}$ получим: $K_{ус} = 4,6875$, $K_{ус макс} = 9,375$, $t_{уст} = 3,690 \cdot 10^{-10} \text{ с}$, $W_{вх} = 1,68 \text{ мм}$, $W_{вых} = 0,76 \text{ мм}$ и $v/c = 0,587$

§ 7.6. Расчет активных фильтров

Активные фильтры (АФ) широко применяются в технике связи и в измерительной технике. Они строятся на основе RC цепей и на активных (усилительных) элементах в виде интегральных операционных усилителей. Отсутствие в АФ громоздких индуктивностей обеспечивает их малые габариты и массу, высокую надежность и технологичность.

Сложные фильтры проектируются в виде последовательного (каскадного) соединения элементарных фильтров второго порядка. В зависимости от характеристик элементарных фильтров могут быть получены фильтры с различными АЧХ и ФЧХ, аппроксимируемыми по Баттерворту, Бесселю, Чебышеву и т. д. [23, 35]. Каскадированию элементарных фильтров и выбору их параметров посвящена обширная справочная литература, причем большинство данных для этого табулировано.

В связи с отмеченным, для разработчиков фильтров представляет интерес расчет на ЭВМ функционально полного комплекта элементарных фильтров второго порядка. Такой расчет для каждого фильтра прихо-

дится, как правило, выполнять неоднократно при различных сочетаниях исходных данных с тем, чтобы получить приемлемые значения расчетных параметров. Таким образом, обеспечивается многовариантный расчет и выбор оптимального варианта фильтра. Методика и алгоритмы подобного расчета подробно описаны в [23].

Ниже приводится существенно усовершенствованная, в сравнении с описанной в [23], система автоматического проектирования электрических схем элементарных АФ. Система состоит из 24 взаимосвязанных программ, обеспечивающих расчет всех 23 элементарных схем АФ. В отличие от [23] система обеспечивает:

1. Информацию пользователю о классификации фильтров и их «меню» (программа 7.33), а также о виде АЧХ и ФЧХ.

2. Автоматический поиск выбранного варианта фильтра (все программы 7.34—7.56 записаны на одной кассете и пронумерованы кодами от 1 до 23). Расположение программ на магнитной ленте, т. е. в ПЗУ, а не в ОЗУ ЭВМ, обеспечивает разгрузку ОЗУ, позволяет увеличить объем информации, предоставляемой пользователю, уменьшает время ввода программы и увеличивает надежность программного поиска.

3. Выдачу стилизованной принципиальной схемы выбранного варианта фильтра с обозначениями ее элементов и выдачу уравнения для операторной функции передачи $T(s)$ (как и в [23] оператор $j\omega$ обозначен буквой s).

Программа 7.33.

```

0010 PRINT'КОМПЛЕКС ПРОГРАММ ДЛЯ АВТОМАТИЧЕСКОГО'
0020 PRINT'ПРОЕКТИРОВАНИЯ АКТИВНЫХ ФИЛЬТРОВ.'
0030 PRINT'РАССЧИТЫВАЮТСЯ СЛЕДУЮЩИЕ ВИДЫ ФИЛЬТРОВ.'
0040 PRINT'НИЗКОЧАСТОТНЫЕ'
0050 PRINT'      Ар ДЕ (А-КОЭФФИЦИЕНТ УСИЛЕНИЯ)  '
0060 PRINT'      !                                     (QP=2)  '
0070 PRINT'      !                                     **      '
0080 PRINT'      К !*****! *      '
0090 PRINT'      ! ! *      '
0100 PRINT'      ! ! *      '
0110 PRINT'      !-----!-----*----- LGT W  '
0120 PRINT'      WM WF      '
0130 PRINT'      F(W) (F-ФАЗА)      '
0140 PRINT'      !      '
0150 PRINT'      0 !*****      '
0160 PRINT'      ! *      '
0170 PRINT'      -90 !-----*      '
0180 PRINT'      ! ! *****      '
0190 PRINT'      -180 !-----!----- LGT W  '
0200 PRINT'      WF      '
0210 PRINT''
0220 INPUT X
0230 PRINT''
0240 PRINT'ВЫСОКОЧАСТОТНЫЕ'
0250 PRINT'      Ар ДЕ (А-КОЭФФИЦИЕНТ УСИЛЕНИЯ)  '
0260 PRINT'      !                                     (QP=2)  '
0270 PRINT'      !                                     **      '
0280 PRINT'      ! *! *      '
0290 PRINT'      К !-----!-----*****      '
0300 PRINT'      ! * !      '
0310 PRINT'      ! * !      '
0320 PRINT'      ! * !      '
0330 PRINT'      *-----!----- LGT W  '

```

```

0340 PRINT' WP
0350 PRINT' F(W) (F-ФАЗА)
0360 PRINT' !
0370 PRINT' 180 !*****
0380 PRINT' ! *
0390 PRINT' 90 !-----*
0400 PRINT' ! !*
0410 PRINT' 0 !-----!-*****----- LGT W
0420 PRINT' WP
0430 PRINT'
0440 INPUT X
0450 PRINT'
0460 PRINT'
0470 PRINT' ПОЛОСНО-ПРОПУСКАЮЩЕ'
0480 PRINT' А, ДЕ (А-КОЭФФИЦИЕНТ УСИЛЕНИЯ)
0490 PRINT' ! (QF=10)
0500 PRINT' К !-----*
0510 PRINT' ! *!*
0520 PRINT' ! *!*
0530 PRINT' ! *!!!*
0540 PRINT' ! *!!! *
0550 PRINT' ! *!!! *
0560 PRINT' !*!!! *
0570 PRINT' *-----!-----*----- LGT W
0580 PRINT' W1 WP W2
0590 PRINT' F(W) (F-ФАЗА)
0600 PRINT' !
0610 PRINT' 90 !*****
0620 PRINT' ! *
0630 PRINT' 0 !-----*
0640 PRINT' ! *
0650 PRINT' -90 !-----!-*****----- LGT W
0660 PRINT' WP
0670 INPUT X
0680 PRINT'
0690 PRINT'
0700 PRINT' ПОЛОСНО-ЗАГРЯЖДАЮЩЕ'
0710 PRINT' А, ДЕ (А-КОЭФФИЦИЕНТ УСИЛЕНИЯ)
0720 PRINT' !
0730 PRINT' К !***** *****
0740 PRINT' ! * *
0750 PRINT' ! *
0760 PRINT' К*QF/QZ !-----*
0770 PRINT' ! !
0780 PRINT' !-----!----- LGT W
0790 PRINT' WP
0800 PRINT' F(W) (F-ФАЗА)
0810 PRINT' 90 !
0820 PRINT' !
0830 PRINT' ! *
0840 PRINT' ! * *
0850 PRINT' 0 !***** * *****
0860 PRINT' ! * *!
0870 PRINT' ! * !
0880 PRINT' ! !
0890 PRINT' -90 !-----!----- LGT W
0900 PRINT' WP
0910 PRINT' * INPUT X
0920 PRINT'

```



```

0930 PRINT'ВСЕПРОПУСКАЮЩИЕ'
0940 PRINT' А, ДЕ (А-КОЭФФИЦИЕНТ УСИЛЕНИЯ) '
0950 PRINT' ! '
0960 PRINT' К !***** '
0970 PRINT' ! '
0980 PRINT' ! '
0990 PRINT' !----- LGT W '
1000 PRINT' F(W) (F-ФАЗА) '
1010 PRINT' ! '
1020 PRINT' 0 !***** '
1030 PRINT' ! * '
1040 PRINT' -180 !-----* '
1050 PRINT' ! !* '
1060 PRINT' -360 !-----!***** LGT W '
1070 PRINT' WP=WZ '
1080 PRINT' '
1090 PRINT' '
1100 PRINT'В ПРИВЕДЕННОМ НИЖЕ СПИСКЕ'
1110 PRINT'ВЫБЕРИТЕ ТИП ФИЛЬТРА. НАБЕРИТЕ LOAD(N)'
1120 PRINT'ГДЕ (N)-НОМЕР ФИЛЬТРА В АПОСТРОФАХ.'
1130 PRINT'НАЖМИТЕ КЛАВИШУ РС': INPUT X
1140 PRINT' '
1150 PRINT' '
1160 PRINT'СПИСОК ФИЛЬТРОВ'
1170 PRINT'НИЗКОДОБОРОТНЫЕ'
1180 PRINT'1. НИЗКОЧАСТОТНЫЙ'
1190 PRINT'2. ПОЛОСНО-ПРОПУСКАЮЩИЙ С РЕЗИСТИВНЫМ ВХОДОМ'
1200 PRINT'3. ПОЛОСНО-ПРОПУСКАЮЩИЙ С ЕМКОСТНЫМ ВХОДОМ'
1210 PRINT'4. ВЫСОКОЧАСТОТНЫЙ'
1220 PRINT'5А. ВСЕПРОПУСКАЮЩИЙ-ДОБОРОТНОСТЬ 0.5 ПОЛОЖИТЕЛЬНАЯ'
1230 PRINT'5Б. ВСЕПРОПУСКАЮЩИЙ-ДОБОРОТНОСТЬ 0.5 ОТРИЦАТЕЛЬНАЯ'
1240 PRINT'6. ВСЕПРОПУСКАЮЩИЙ'
1250 PRINT'7. ПОЛОСНО-ЗАГРАЖДАЮЩИЙ'
1260 PRINT' '
1270 PRINT'СРЕДНЕДОБОРОТНЫЕ'
1280 PRINT'8. НИЗКОЧАСТОТНЫЙ'
1290 PRINT'9. ПОЛОСНО-ПРОПУСКАЮЩИЙ С РЕЗИСТИВНЫМ ВХОДОМ'
1300 PRINT'10 ПОЛОСНО-ПРОПУСКАЮЩИЙ С ЕМКОСТНЫМ ВХОДОМ'
1310 PRINT'11. ВЫСОКОЧАСТОТНЫЙ'
1320 PRINT'12. ВСЕПРОПУСКАЮЩИЙ'
1330 PRINT'13. ПОЛОСНО-ЗАГРАЖДАЮЩИЙ'
1340 PRINT'14. НЧ И ВЧ С НУЛЕМ ПЕРЕДАЧИ'
1350 PRINT'15. ПОЛОСНО-ЗАГРАЖДАЮЩИЙ НЧ И ВЧ С НУЛЕМ ПЕРЕДАЧИ'
1360 INPUT X
1370 PRINT' '
1380 PRINT' '
1390 PRINT'ВЫСОКОДОБОРОТНЫЕ'
1400 PRINT'16. НИЗКОЧАСТОТНЫЙ'
1410 PRINT'17. ПОЛОСНО-ПРОПУСКАЮЩИЙ'
1420 PRINT'18. ВЫСОКОЧАСТОТНЫЙ'
1430 PRINT'19. ВСЕПРОПУСКАЮЩИЙ'
1440 PRINT'20. ПОЛОСНО-ЗАГРАЖДАЮЩИЙ'
1450 PRINT'21. НЧ И ВЧ С НУЛЕМ ПЕРЕДАЧИ'
1460 PRINT'22. УНИВЕРСАЛЬНЫЙ. СХЕМА 1'
1470 PRINT'23. УНИВЕРСАЛЬНЫЙ. СХЕМА 2'
1480 PRINT' '
1490 PRINT' '
1500 PRINT' '
1510 PRINT' '
1520 PRINT'ПРИ НЕОБХОДИМОСТИ ПОВТОРНОГО ПРОСМОТРА СПИСКА'
1530 PRINT'НАБЕРИТЕ GOTO 1160'
1540 PRINT' '
1550 PRINT' '
1560 PRINT' '
1570 PRINT' '
1580 PRINT' '
1590 PRINT' ':STOP
1600 PRINT' '
1610 PRINT' '
1620 END

```

Принципиальные схемы фильтров приведены в поле операторов PRINT в начале каждой программы. Поэтому отдельно в виде рисунков они не приводятся. При расчетах частоты задаются в герцах, сопротивления — в омах и емкости конденсаторов — в фарадах. Необязательный параметр (т. е. не влияющий на АЧХ и ФЧХ, но влияющий на номиналы расчетных резисторов и конденсаторов) в примерах помечен знаком *. Добротность $q_p = Q$ однозначно определяет характер переходных процессов (апериодический, критический или колебательный). Обычно достаточно знать передаточную функцию в операторном виде. Однако если пользователь желает провести расчеты АЧХ, ФЧХ или ПХ по заданной передаточной функции, он может воспользоваться формулами для основных параметров фильтров, приводимыми перед каждой программой. Общий объем памяти, занимаемый всеми 24 программами расчета АФ, составляет около 0,3 Мбайт.

Низкодобротные фильтры ($Q = q_p \leq 2$).

1. Низкочастотный.

Передаточная функция $T(s)$: $T(s) = K[\omega_p^2 / (s^2 + (\omega_p/q_p)s + \omega_p^2)]$,
 $R_1 = R_{11}R_{12}/(R_{11} + R_{12})$, $\omega_p = 1/\sqrt{R_1C_2R_3C_4}$,
 $q_p = \sqrt{R_3C_2/R_1C_4}/(1 + R_3/R_7)$.

Постоянный коэффициент усиления K :
 $K = R_{12}/(R_{11} + R_{12})$.

Произведение усиление — чувствительность (ПУЧ): $\text{ПУЧ} = q_p \sqrt{C_2R_1/C_4R_3}$.

Программа 7.34.

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА НЧ НИЗКОДОБРОТНОГО'
02 PRINT '-----'
03 PRINT '      !'
04 PRINT '      C2 ===          ОУ !'
05 PRINT '      !          R3          !'
06 PRINT ' <-!====!*!====!*-----!+ ! !'
07 PRINT '      R11          !          !*->'
08 PRINT '      -          !          !'
09 PRINT '      !!          C4 === ! ! ! !'
10 PRINT '      R12 !!          ! ! ! !'
11 PRINT '      -          !          !'
12 PRINT '      !          !          !'
13 PRINT '      ---          !          !'
20 INPUT 'FP='F: INPUT 'QP='Q
30 INPUT 'C2='C2 : INPUT 'C4='C4
40 IF C2=4*Q^2*C4 THEN GO
50 PRINT 'ФИЛЬТР НЕ РЕАЛИЗУЕМ.'
55 PRINT 'ВВЕДИТЕ НОВЫЕ C2 И C4: 60TO 30
60 LET A=C2/(2*Q^2*C4)-1
70 LET P=A+SQR(A^2-1)
80 LET R1=1/(2*P*F*SQR(P*C2*C4))
90 LET R3=P*R1 : LET K=Q*SQR(C2/P/C4)
100 INPUT 'KBX='K1 : IF K1=0 THEN 140
110 IF 1-K1<=0 THEN 140
120 LET K=K1 * LET R8=R1/K
130 LET R9=R1/(1-K) : 60TO 150
140 LET R8=R1 : LET R9=9E99 : LET K=1
150 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА '
160 PRINT 'R1='R8 : PRINT 'R12='R9
170 PRINT 'R3='R3 : PRINT 'K='K
180 PRINT 'ПУЧ='X
190 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
200 PRINT 'T(S)=K*(WP^2)/(S^2+(WP/Q)*S+(WP^2))'
210 PRINT: 60TO 100: END

```

Пример А. Вводимые параметры: $F = 2,3 \cdot 10^3$, $Q = 1,2$, $C_2 = 27 \cdot 10^{-9}$, $C_4 = 3,3 \cdot 10^{-9}$, $K_{\text{вх}} = 600 \cdot 10^{-3}$. Результаты расчета: $R_{11} =$

$= 6,6393365 \cdot 10^3$, $R_{12} = 9,9590048 \cdot 10^3$, $R_3 = 13,49059 \cdot 10^3$, ПУЧ = 1,8652139.

Пример В. Вводимые параметры: $F = 2,3 \cdot 10^3$, $Q = 1,2$, $R_{12} = 1 \cdot 10^{90}$, $C_2 = 27 \cdot 10^{-9}$, $C_4 = 3,3 \cdot 10^{-9}$, $K_{\text{вх}} = 1$. Результаты расчета: $R_{11} = 3,9836019 \cdot 10^3$, $R_{12} = 1 \cdot 10^{90}$, $R_3 = 13,49059 \cdot 10^3$, ПУЧ = 1,8652139.

2. Полосно-пропускающий с резистивным входом.

$T(s) = -K \{ (\omega_p/q_p) s / [s^2 + (\omega_p/q_p) s + \omega_p^2] \}$,

$R_1 = R_1R_{12}/(R_1 + R_{12})$, $\omega_p = 1/\sqrt{R_1C_2C_3R_4}$.

$q_p = \sqrt{R_4C_2/R_1C_3}/(1 + C_2/C_3)$,

$K = R_{12}K_0/(R_{11} + R_{12})$,

$K_0 = \text{ПУЧ} = q_p^2 [1 + (C_3/C_2)]$.

Программа 7.35.

Пример А. Вводимые параметры: $F = 2,3 \cdot 10^3$, $Q = 1,4$, $C_2 = 68 \cdot 10^{-9}$, $C_3 = 15 \cdot 10^{-9}$, $K_{\text{вх}} = 1,5$. Результаты расчета: $R_{11} = 949,77374$, $R_{12} = 1,5965215 \cdot 10^3$, $R_4 = 7,8831221 \cdot 10^3$, $K = 1,5$, ПУЧ = 2,3923529.

Пример В. Вводимые параметры: $F = 2,3 \cdot 10^3$, $Q = 1,4$, $C_2 = 68 \cdot 10^{-9}$, $C_3 = 15 \cdot 10^{-9}$, $R_{12} = 1 \cdot 10^{90}$, $K_{\text{вх}} = 0$. Результаты расчета: $R_{11} = 595,50604$, $R_{12} = 1 \cdot 10^{90}$, $R_4 = 7,8831221 \cdot 10^3$, $K = 2,3923529$. ПУЧ = 2,3923529.

3. Полосно-пропускающий с емкостным входом.

$T(s) = -K \{ (\omega_p/q_p) s / [s^2 + (\omega_p/q_p) s + \omega_p^2] \}$,

$C_1 = C_{11} + C_{12}$, $\omega_p = 1/\sqrt{C_1R_2R_3C_4}$,

4. Высокочастотный.

$$T(s) = Ks^2 / [s^2 + (\omega_p/q_p)s + \omega_p^2], \quad C_{11} = C_{11} + C_{12}, \quad \omega_p = 1/\sqrt{C_1 R_2 C_3 R_4}, \quad q_p = \sqrt{R_4 C_1 / R_2 C_3} / (1 + C_1 / C_3), \quad K = C_{11} / (C_{11} + C_{12}), \quad \text{ПУЧ} = q_p^2 [1 + (C_3 / C_1)].$$

Программа 7.37.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА ВЧ НИЗКОДОБОРНОГО'
02 PRINT'
03 PRINT'
04 PRINT'
05 PRINT'
06 PRINT'
07 PRINT'
08 PRINT'
09 PRINT'
10 PRINT'
11 PRINT'
12 PRINT'
13 PRINT'
14 PRINT'
15 PRINT'
16 PRINT'
17 PRINT'
20 INPUT'FP='F: INPUT'QP='Q
30 INPUT'C3='C3: INPUT'C11='C8: INPUT'C12='C9
40 LETC1=C8+C9: LETP=Q^2*(2+(C3/C1)+(C1/C3))
50 LETR2=1/(2*PI*I*F*SQR(P*C1*C3)): LETR4=P*R2
60 LETX=Q*SQR(P*C3/C1): LETK=C8/C1
70 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА'
80 PRINT'R2='R2: PRINT'R4='R4: PRINT'K='K
90 PRINT'ПУЧ='X
100 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
110 PRINT'T(S)=K*(S^2)/(S^2+(WP/Q)*S+WP^2)'
120 END

```

Пример А. Вводимые параметры: $F=2,5 \cdot 10^3$, $Q=1,5$, $C_{11}=100 \cdot 10^{-9}$, $C_{12}=15 \cdot 10^{-9}$, $C_3=15 \cdot 10^{-9}$. Результаты расчета: $R_2=326,47168$, $R_4=7,1965712 \cdot 10^3$, $K=869,56522 \cdot 10^{-3}$, ПУЧ=2,5434783.

Пример В. Вводимые параметры: $F=2,5 \cdot 10^3$, $Q=1,5$, $C_{11}=27 \cdot 10^{-9}$, $C_{12}=0$, $C_3=15 \cdot 10^{-9}$, $K=1$. Результаты расчета: $R_2=1,0105076 \cdot 10^3$, $R_4=9,9029742 \cdot 10^3$, $K=1$, ПУЧ=3,5.

5. а) всепропускающий $Q=0,5$; б) всепропускающий $Q=-0,5$.

$$T_p(s) = [s^2 - (\omega_p/q_p)s + \omega_p^2] / [s^2 + (\omega_p/q_p)s + \omega_p^2],$$

$$T_N(s) = -[s^2 - (\omega_p/q_p)s + \omega_p^2] / [s^2 + (\omega_p/q_p)s + \omega_p^2],$$

$$R_s = R_4 + R_5, \quad C_s = C_1 C_2 / (C_1 + C_2),$$

$$\omega_p = 1/\sqrt{R_4 R_5 C_3 C_5} = 1/\sqrt{R_5 R_6 C_1 C_2},$$

$$q_p = 1/\omega_p (R_4 C_3 + R_5 C_2).$$

Программа 7.38.

Пример А. Вводимые параметры: $F=2,5 \cdot 10^3$, $Q=300 \cdot 10^{-3}$, $C_1=68 \cdot 10^{-9}$, $C_2=6,8 \cdot 10^{-9}$, $C_3=15 \cdot 10^{-9}$. Результаты расчета: $R_4=8,0381583 \cdot 10^3$, $R_5=5,4374616 \cdot 10^3$, $R_6=650,41967$, $R_7=1,6960889 \cdot 10^3$, $R_8=25,76939 \cdot 10^3$, $R_9=319,1209$.

Пример В. Вводимые параметры: $F=2,5 \cdot 10^3$, $Q=300 \cdot 10^{-3}$, $C_1=27 \cdot 10^{-9}$, $C_2=6,8 \cdot 10^{-9}$, $C_3=27 \cdot 10^{-9}$. Результаты расчета: $R_4=5,2115421 \cdot 10^3$, $R_5=5,3024215 \cdot 10^3$, $R_6=2,0995252 \cdot 10^3$, $R_7=1,0667594 \cdot 10^3$, $R_8=25,90443 \cdot 10^3$, $R_9=818,44118$.

6. Всепропускающий. $T(s) = K[s^2 - (\omega_p/q_p)s + \omega_p^2] / [s^2 + (\omega_p/q_p)s + \omega_p^2]$,

$$\omega_p = 1/\sqrt{R_1 C_2 C_3 R_4}, \quad 2[(1/R_4 C_2) + (1/R_4 C_3)] = (1/R_1 C_2) (R_5/R_6),$$

$$q_p = \sqrt{R_4/R_1} / (\sqrt{C_2/C_3} + \sqrt{C_3/C_2}), \quad K = R_6 / (R_5 + R_6), \quad \text{ПУЧ} = q_p \sqrt{R_4 C_3 / R_1 C_2}.$$

Примечание. Начальные условия $C_2=C_3$ или $C_2 > C_3$ для низких значений ПУЧ.

Программа 7.39.

Пример А. Вводимые параметры: $F=2,3 \cdot 10^3$, $Q=1,6$, $C_2=10 \cdot 10^{-9}$, $C_3=10 \cdot 10^{-9}$, $R_6^0=6,8 \cdot 10^3$. Результаты расчета: $R_1=2,1624313 \cdot 10^3$, $R_4=22,143296 \cdot 10^3$, $R_5=2,65625 \cdot 10^3$, $K=719,10112 \cdot 10^{-3}$, ПУЧ=5,12.

Пример В. Вводимые параметры: $F=2,3 \cdot 10^3$, $Q=1,6$, $C_2=22 \cdot 10^{-9}$, $C_3=10 \cdot 10^{-9}$, $R_6^0=0$. Результаты расчета: $R_1=1,3515196 \cdot 10^3$, $R_4=16,104216 \cdot 10^3$, $R_5=5,3710938 \cdot 10^3$, $C_3=650,57179 \cdot 10^{-3}$, $R_6^0=3,7236364$.

Если значение R_6 задано равным 0, программа присваивает ему подходящее значение $R_6=1 \cdot 10^4$ Ом.

7. Полосно-заграждающий.

$$T(s) = K(s^2 + \omega_p^2) / [s^2 + (\omega_p/q_p)s + \omega_p^2],$$

$$\omega_p = 1/\sqrt{R_1 C_2 C_3 R_4}, \quad (1/R_4 C_2) + (1/R_4 C_3) = (1/R_1 C_2) (R_5/R_6),$$

$$q_p = \sqrt{R_4/R_1} / (\sqrt{C_2/C_3} + \sqrt{C_3/C_2}), \quad K = R_6 / (R_5 + R_6), \quad \text{ПУЧ} = q_p \sqrt{R_4 C_3 / R_1 C_2}.$$

Программа 7.40.

Пример А. Вводимые параметры: $F=2,2 \cdot 10^3$, $Q=1,7$, $C_2=15 \cdot 10^{-9}$, $C_3=100 \cdot 10^{-9}$, $R_6^0=22 \cdot 10^3$. Результаты расчета: $R_1=370,04172$, $R_4=9,4287247 \cdot 10^3$, $R_5=992,92914$, $K=956,81589 \cdot 10^{-3}$, ПУЧ=22,156667.

Программа 7.38.

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ВСЕПРОПУСКАЮЩЕГО С Q=+0.5'
02 PRINT 'ВАРИАНТ 1'
03 PRINT '
04 PRINT '
05 PRINT '
06 PRINT '
07 PRINT '
08 PRINT '
09 PRINT '
10 PRINT '
11 PRINT '
12 PRINT '
13 PRINT '
14 PRINT '
15 PRINT '
16 PRINT '
17 PRINT '
18 PRINT '
19 PRINT '
20 PRINT '
21 PRINT 'РАСЧЕТ ФИЛЬТРА ВСЕПРОПУСКАЮЩЕГО С Q=-0.5'
22 PRINT 'ВАРИАНТ 2'
23 PRINT '
24 PRINT '
25 PRINT '
26 PRINT '
27 PRINT '
28 PRINT '
29 PRINT '
30 PRINT '
31 PRINT '
32 PRINT '
33 PRINT '
34 PRINT '
35 PRINT '
36 PRINT '
37 PRINT '
38 PRINT '
39 PRINT '
40 PRINT '
50 INPUT 'FP='F: INPUT 'QP='Q
60 INPUT 'C1='C1: INPUT 'C2='C2: INPUT 'C3='C3
70 LET H=1-((C1+C2)*(C2+C3)*4*Q^2)/C1/C3
80 IF H>0 THEN 110
90 PRINT 'ФИЛЬТР НЕ РЕАЛИЗУЕМ. ВВЕДИТЕ НОВЫЕ C1,C2,C3.' :GOTO 100
100 PRINT 'C1=C3>=C2*2*Q/(1-2*Q)':GOTO 60
110 LET R4=(1+SQR(H))/(2*2*PI*F*Q*(C2+C3))
120 LET R5=(C1+C2)/(R4*(2*PI*F)^2*C1*C2*C3)
130 LET R6=1/((2*PI*F)^2*C1*C2*(R4+R5))
140 LET R7=(1-SQR(H))/(2*2*PI*F*Q*(C2+C3))
150 LET R8=(C1+C2)/(R7*(2*PI*F)^2*C1*C2*C3)
160 LET R9=1/((2*PI*F)^2*C1*C2*(R7+R8))
170 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА.'
180 PRINT 'R4='R4: PRINT 'R5='R5: PRINT 'R6='R6
190 PRINT 'R7='R7: PRINT 'R8='R8: PRINT 'R9='R9
200 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ДЛЯ ФИЛЬТРА С Q=+0.5'
210 PRINT 'T(S)=(S^2-(WP/Q)*S+WP^2)/(S^2+(WP/Q)*S+WP^2)'
220 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ДЛЯ ФИЛЬТРА С Q=-0.5'
230 PRINT 'T(S)=-((S^2-(WP/Q)*S+WP^2)/(S^2+(WP/Q)*S+WP^2))'
240 END

```

: INPUT X

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ВСЕПРОПУСКАЮЩЕГО НИЗКОДОБОРНОГО'
02 PRINT '
03 PRINT '
04 PRINT '
05 PRINT '
06 PRINT '
07 PRINT '
08 PRINT '
09 PRINT '
10 PRINT '
11 PRINT '
12 PRINT '
13 PRINT '
14 PRINT '
15 PRINT '
16 PRINT '
17 PRINT '
18 PRINT '
20 INPUT 'FP=' F: INPUT 'QP=' Q
30 INPUT 'C2=' C2: INPUT 'C3=' C3: INPUT 'R6=' R6
40 LETP=Q^2*(2+(C2/C3)+(C3/C2))
50 LETR1=1/(2*PI*F*SQR(P*C2*C3))
60 LETR4=P*R1: LETX=Q*SQR(P*C3/C2)
70 IF R6=0 THEN 100
80 LETR5=R6*(1+(C2/C3))*2/P
90 LETK=R6/(R5+R6): GOTO 110
100 LETR6=1E4
110 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА'
120 PRINT 'R1=' R1: PRINT 'R4=' R4
130 PRINT 'R5=' R5: PRINT 'K=' K
140 PRINT 'ПУЧ=' X
150 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
160 PRINT 'T(S)=K*(S^2-(WP/Q)*S+WP^2)/(S^2+(WP/Q)*S+WP^2)'
170 END

```

Программа 7.40.

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ПОЛОСНО-ЗАГРЯЖДАЮЩЕГО НИЗКОДОБОРНОГО'
02 PRINT '
03 PRINT '
04 PRINT '
05 PRINT '
06 PRINT '
07 PRINT '
08 PRINT '
09 PRINT '
10 PRINT '
11 PRINT '
12 PRINT '
13 PRINT '
14 PRINT '
15 PRINT '
16 PRINT '
17 PRINT '
18 PRINT '
20 INPUT 'FP=' F: INPUT 'QP=' Q
30 INPUT 'C2=' C2: INPUT 'C3=' C3: INPUT 'R6=' R6
40 LETP=Q^2*(2+(C2/C3)+(C3/C2))
50 LETR1=1/(2*PI*F*SQR(P*C2*C3))
60 LETR4=P*R1: LETX=Q*SQR(P*C3/C2)
70 IF R6=0 THEN 100
80 LETR5=R6*(1+(C2/C3))/P
90 LETK=R6/(R5+R6): GOTO 110
100 LETR6=1E4
110 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА'
120 PRINT 'R1=' R1: PRINT 'R4=' R4
130 PRINT 'R5=' R5: PRINT 'K=' K
140 PRINT 'ПУЧ=' X
150 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
160 PRINT 'T(S)=K*(S^2+WP^2)/(S^2+(WP/Q)*S+WP^2)'
170 END

```



```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ПОЛ.-ПРОПУСК. СРЕДНЕДОБОТН. С R-ВХОДОМ'
02 PRINT '
03 PRINT '
04 PRINT '
05 PRINT '
06 PRINT '
07 PRINT '
08 PRINT '
09 PRINT '
10 PRINT '
11 PRINT '
12 PRINT '
13 PRINT '
14 PRINT '
15 PRINT '
16 PRINT '
17 PRINT '
18 PRINT '
19 PRINT '
20 PRINT '
21 PRINT '
30 INPUT 'FP=' F: INPUT 'QP=' Q
40 INPUT 'C2=' C2: INPUT 'C3=' C3: INPUT 'R6=' R6
50 LET P1=C2/C3/(4*(Q^2))*((SQR(1+12*(Q^2))*(1+(C3/C2)))-1)^2
60 PRINT 'P ДЛЯ ОПТИМАЛЬНОГО ПУЧ=P1
70 PRINT 'ВВЕДИТЕ КОД 0, ЕСЛИ P=РОПТ'
80 INPUT 'ВВЕДИТЕ КОД 1, ЕСЛИ P ДРУГОЕ' A
90 IF A=0 THEN 120
100 IF A<>0 THEN 110
110 LET P=P1:GOTO 130
120 INPUT 'ВВЕДИТЕ P' P
130 LET R1=1/(2*PI*F*SQR(P*C2*C3)): LET R4=P*R1
140 IF R6=0 THEN LET R6=1E4
150 LET R5=R6*((1/P)*(1+(C2/C3))-(SQR(C2/P/C3))*(1/Q))
160 LET K0=Q*(1+(R5/R6))*SQR(P*C3/C2)
170 LET X=(1+(R5/R6))*K0
180 INPUT 'КВХ=' K1: IF K1=0 THEN 200
190 IF (K0-K1)>0 THEN 210
200 LET R8=R1: LET R9=1E90: LET K2=K0:GOTO 220
210 LET K2=K1: LET R8=(K0/K1)*R1: LET R9=(K0/(K0-K1))*R1
220 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА.'
230 PRINT 'R11=' R8: PRINT 'R12=' R9: PRINT 'R4=' R4
240 PRINT 'R5=' R5: PRINT 'K=' K2: PRINT 'P=' P
250 PRINT 'ПУЧ=' X
260 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
270 PRINT 'T(S)=-K*(WP/Q)*S/(S^2+(WP/Q)*S+WP^2)'
280 GOTO 120:END
    
```

Пример В. Вводимые параметры: $F=2,3 \cdot 10^3$, $Q=6$, $C_2=15 \cdot 10^{-9}$, $C_3=3,9 \cdot 10^{-9}$, $R_6=0$. Результаты расчета: $P=13,34444$, $\text{ПУЧ}=18,130512$, $R_{11}=2,4766475 \cdot 10^3$, $R_4=33,049473$, $R_{12}=1 \cdot 10^{90}$, $R_5=2,7368193$, $K=14,234725$.

Если задано $R_6=0$, программа присваивает R_6 значение $1 \cdot 10^4$ Ом. Настройка: (1) f_p резистором R_4 , (2) q_p резистором R_5 .

10. Полосно-пропускающий с емкостным входом.

$$T(s) = -K(\omega_p/q_p) s / [s^2 + (\omega_p/q_p) s + \omega_p^2],$$

$$C_1 = C_{11} + C_{12},$$

$$\omega_p = 1/\sqrt{C_1 R_2 R_3 C_4},$$

$$q_p = \sqrt{C_1 R_3 / C_4 R_2} / [1 + (R_3/R_2) - (C_1 R_5 / C_4 R_6)],$$

$$K = [C_{11} / (C_{11} + C_{12})] q_p [1 + (R_5/R_6)] \sqrt{C_1 R_2 / C_4 R_3},$$

$$\text{ПУЧ} = q_p [1 + (R_5/R_6)]^2 \sqrt{C_1 R_2 / C_4 R_3}.$$

Программа 7.43.

Пример А. Вводимые параметры: $F=2,1 \cdot 10^3$, $Q=5$, $C_{11}=68 \cdot 10^{-9}$, $C_{12}=$

$=15 \cdot 10^{-9}$, $C_4=5,1 \cdot 10^{-9}$, $R_6^s=15 \cdot 10^3$. Результаты расчета: $P=6,4407124$, $\text{ПУЧ}=14,08843$, $R_2=1,4514756 \cdot 10^8$, $R_3=9,3485371 \cdot 10^3$, $R_5=4,970735 \cdot 10^3$, $K=8,6694315$.

Пример В. Вводимые параметры: $F=2,1 \cdot 10^3$, $Q=5$, $C_{11}=27 \cdot 10^{-9}$, $C_{12}=0$, $C_4=4,7 \cdot 10^{-9}$, $R_6=0$. Результаты расчета: $P=2,500919$, $\text{ПУЧ}=16,541822$, $R_2=4,2542227 \cdot 10^3$, $R_3=10,639466 \cdot 10^3$, $R_5=4,7745784 \cdot 10^3$, $K=11,196135$.

Если задано $R_6=0$, программа задает $R_6=1 \cdot 10^4$ Ом. Настройка: (1) f_p резистором R_2 или R_3 , (2) q_p резистором R_5 .

11. Высокочастотный.

$$T(s) = K s^2 / [s^2 + (\omega_p/q_p) s + \omega_p^2],$$

$$C_1 = C_{11} + C_{12},$$

$$\omega_p = 1/\sqrt{C_1 R_2 C_3 R_4}, \quad q_p = \sqrt{R_4 C_1 / R_2 C_3} / [1 + (C_1/C_3) - (R_4 R_6 / R_2 R_5)],$$

$$K = (C_{11}/C_1) [1 + (R_6/R_5)], \quad \text{ПУЧ} = q_p [1 + (R_6/R_5)]^2 \sqrt{R_4 C_3 / R_2 C_1}.$$


```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ПОЛ.-ПРОПУСК. СРЕДНЕДОБОТНО. С С-ВХОДОМ'
02 PRINT'
03 PRINT'
04 PRINT'
05 PRINT'
06 PRINT'
07 PRINT'
08 PRINT'
09 PRINT'
10 PRINT'
11 PRINT'
12 PRINT'
13 PRINT'
14 PRINT'
15 PRINT'
16 PRINT'
17 PRINT'
18 PRINT'
19 PRINT'
20 PRINT'
21 PRINT'
30 INPUT 'FP=' F: INPUT 'QP=' Q: INPUT 'C11=' C8
40 INPUT 'C12=' C9: INPUT 'C4=' C4: INPUT 'R6=' R6
50 LET C1=C8+C9
60 LET P1=C1/C4/(36*(Q^2))*((SQR(1+12*(Q^2)*(1+(C4/C1))))+1)^2)
70 PRINT 'P ДЛЯ ОПТИМАЛЬНОГО ПУЧ' P1
80 PRINT 'ВВЕДИТЕ КОД 0, ЕСЛИ P=РОПТ'
90 PRINT 'ВВЕДИТЕ КОД 1, ЕСЛИ P ДРУГОЕ' A
100 IF A=0 THEN 130
110 IF A<>0 THEN 120
120 LET P=P1:GOTO 140
130 INPUT 'ВВЕДИТЕ P' P
140 LET R2=1/(2*PI*F*SQR(P*C1*C4)):LETR3=P*R2
150 IF R6=0 THEN LET R6=1E4
160 LET R5=R6*((C4/C1)*(1+P)-(1/Q)*SQR(P*C4/C1))
170 LET K=(C8/C1)*(1+(R5/R6))*Q*SQR(C1/P/C4)
180 LET X=Q*((1+(R5/R6))^2)*SQR(C1/P/C4)
190 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА.'
200 PRINT 'R2=' R2:PRINT 'R3=' R3:PRINT 'R5=' R5
210 PRINT 'K=' K:PRINT 'P=' P:PRINT 'ПУЧ=' X
220 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
230 PRINT 'T(S)=-K*(WP/Q)*S/(S^2+(WP/Q)*S+WP^2)'
240 GOTO 130:END

```

Программа 7.44.

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ВЧ СРЕДНЕДОБОТНОГО'
02 PRINT'
03 PRINT'
04 PRINT'
05 PRINT'
06 PRINT'
07 PRINT'
08 PRINT'
09 PRINT'
10 PRINT'
11 PRINT'
12 PRINT'
13 PRINT'
14 PRINT'
15 PRINT'
16 PRINT'
17 PRINT'
18 PRINT'
19 PRINT'
20 PRINT'
21 PRINT'
30 INPUT 'FP=' F: INPUT 'QP=' Q: INPUT 'C11=' C8
40 INPUT 'C12=' C9: INPUT 'C3=' C3: INPUT 'R5=' R5
50 LET C1=C8+C9
60 LET P1=C1/C3/(4*(Q^2))*((SQR(1+12*(Q^2)*(1+(C3/C1))))-1)^2)

```

```

170 PRINT'ВВЕДИТЕ КОД 0,ЕСЛИ P=РОПТ'
90 INPUT'ВВЕДИТЕ КОД 1,ЕСЛИ P ДРУГОЕ'А
100 IF A=0 THEN 130
110 IFA<>0 THEN 120
120 LETP=P1:GOTO 140
130 INPUT'ВВЕДИТЕ P'P
140 LETR2=1/(2*#PI*F*SQR(P*C1*C3)):LETR4=P*R2
150 IF R5=0 THEN LETR5=1E4
160 LETR6=R5*((1/P)*(1+(C1/C3))-((SQR(C1/P/C3))/Q))
170 LETK=(C8/C1)*(1+(R6/R5))
180 LETX=Q*((1+(R6/R5))^2)*SQR(P*C3/C1)
190 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА.'
200 PRINT'R2='R2:PRINT'R4='R4:PRINT'R6='R6
210 PRINT'K='K:PRINT'P='P:PRINT'ПУЧ='X
220 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
230 PRINT'T(S)=K*(S^2)/(S^2+(W/P/Q)*S+W/P^2)'
240 GOTO 130:END

```

Пример А. Вводимые параметры: $F = 2,2 \cdot 10^3$, $Q = 7$, $C_{11} = 100 \cdot 10^{-9}$, $C_{12} = 15 \cdot 10^{-9}$, $C_3 = 5,1 \cdot 10^{-9}$, $R_5 = 6,8 \cdot 10^3$. Результаты расчета: $P = 65,170699$, ПУЧ = 19,415779, $R_2 = 370,03027$, $R_4 = 24,115131 \cdot 10^3$, $R_6 = 1,8857261 \cdot 10^3$, $K = 1,1107067$.

Пример В. Вводимые параметры: $F = 2,2 \cdot 10^3$, $Q = 7$, $C_{11} = 15 \cdot 10^{-9}$, $C_{12} = 0$, $C_3 = 4,7 \cdot 10^{-9}$, $R_5 = 0$. Результаты расчета: $P = 11,701459$, ПУЧ = 22,083997, $R_2 = 2,5187373 \cdot 10^3$, $R_4 = 29,472902 \cdot 10^3$, $R_6 = 2,8359544 \cdot 10^3$, $K = 1,2835954$.

Если задано $R_5 = 0$, программа присваивает R_5 значение $1 \cdot 10^4$ Ом. Настройка: (1) j_p резистором R_2 или R_4 , (2) q_p резистором R_5 .

12. Всепропускающий.

$$T(s) = [s^2 - (\omega_p/q_p)s + \omega_p^2] / [s^2 + (\omega_p/q_p)s + \omega_p^2]$$

$$R_p = R_4 R_5 / (R_4 + R_5), \quad \omega_p = 1 / \sqrt{R_1 C_2 C_3 R_p}$$

$$q_p = \omega_p / [(1/R_p C_2)(R_7/R_6) - (1/R_1 C_2) - (1/R_1 C_3)] = \omega_p / [(1/R_1 C_2) + (1/R_1 C_3) + (1/R_4 C_2) - (1/R_5 C_2)(R_7/R_6)]$$

$$\text{ПУЧ} = q_p (R_p/R_5) [1 + (R_7/R_6)]^2 \sqrt{R_1 C_3 / R_p C_2}$$

Программа 7.45.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА ВСЕПРОПУСКАЮЩЕГО СРЕДНЕДОБОРНОГО'
02 PRINT'
03 PRINT'
04 PRINT'
05 PRINT'
06 PRINT'
07 PRINT'
08 PRINT'
09 PRINT'
10 PRINT'
11 PRINT'
12 PRINT'
13 PRINT'
14 PRINT'
15 PRINT'
16 PRINT'
17 PRINT'
18 PRINT'
19 PRINT'
20 PRINT'
21 PRINT'
22 PRINT'
23 PRINT'
30 INPUT'FP='F:INPUT'QP='Q
40 INPUT'C2='C2:INPUT'C3='C3:INPUT'R6='R6
50 LETX=C2/C3
60 LETB=6*(Q^2)*((1+X)^2)/(2*(Q^2)*(1+X)-X)
70 LETP1=(1/B)*SQR(1+((B^2)/(3*((1+X)^2))))-1)
80 PRINT'P ДЛЯ ОПТИМАЛЬНОГО ПУЧ'P1
90 PRINT'ВВЕДИТЕ КОД 0,ЕСЛИ P=РОПТ'
100 INPUT'ВВЕДИТЕ КОД 1,ЕСЛИ P ДРУГОЕ'А
110 IF A=0 THEN 140
120 IF A<>0 THEN 130
130 LETP=P1:GOTO 150
140 INPUT'ВВЕДИТЕ P'P
150 LETR1=1/(2*#PI*F*SQR(C2*C3*P))
160 LETR=P*R1
170 IFR6=0 THEN LETR6=1E4
180 LETR7=R6*(P*(1+(C2/C3))+(1/Q)*SQR(P*C2/C3))

```

```

190 LET H=1-(2*(SQR(P*C2/C3))/Q/(1+(R7/R6)))
200 LET R5=R/H:LETR4=R/(1-H)
210 LET Z=H*Q*((1+(R7/R6))^2)*SQR(C3/P/C2)
220 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА.'
230 PRINT 'R1='R1:PRINT 'R4='R4:PRINT 'R5='R5
240 PRINT 'R7='R7:PRINT 'P='P:PRINT 'ПУЧ='Z
250 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
260 PRINT 'T(S)=(S^2-(W/P/Q)*S+W/P^2)/(S^2+(W/P/Q)*S+W/P^2)'
270 GOTO 140:END

```

Пример А. Вводимые параметры:
 $F=2.4 \cdot 10^3$, $Q=5$, $C_2=33 \cdot 10^{-9}$, $C_3=10 \cdot 10^{-9}$,
 $R_6=6.8 \cdot 10^3$. Результаты расчета: $P=$
 $=78,117733 \cdot 10^{-3}$, ПУЧ=16,473246, $R_1=$
 $=13,06102 \cdot 10^3$, $R_4=7,2215149 \cdot 10^3$, $R_5=$
 $=1,1881686 \cdot 10^2$, $R_7=2,9746737 \cdot 10^3$.

Пример В. Вводимые параметры:
 $F=2.4 \cdot 10^3$, $Q=5$, $C_2=18 \cdot 10^{-9}$, $C_3=4.7 \cdot 10^{-9}$,

13. Полосно-заграждающий.

$$T(s) = \frac{(s^2 + \omega_p^2)}{[s^2 + (\omega_p/q_p)s + \omega_p^2]}$$

$$R_p = R_4 R_5 / (R_4 + R_5), \quad \omega_p = 1/\sqrt{R_1 C_2 C_3 R_p}$$

$$(1/R_1 C_3) + (1/R_1 C_3) = (1/R_p C_2) (R_7/R_6)$$

$$q_p = \omega_p / [(1/R_1 C_2) + (1/R_1 C_3) + (1/R_4 C_2) - (1/R_5 C_2) (R_7/R_6)]$$

$$\text{ПУЧ} = q_p (R_p/R_5) [1 + (R_7/R_6)]^2 \sqrt{R_1 C_3 / R_p C_2}$$

Программа 7.46.

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ПОЛОСНО-ЗАГРАЖДАЮЩЕГО СРЕДНЕДОБОРНОГО.'
02 PRINT '
03 PRINT '
04 PRINT '
05 PRINT '
06 PRINT '
07 PRINT '
08 PRINT '
09 PRINT '
10 PRINT '
11 PRINT '
12 PRINT '
13 PRINT '
14 PRINT '
15 PRINT '
16 PRINT '
17 PRINT '
18 PRINT '
19 PRINT '
20 PRINT '
21 PRINT '
30 INPUT 'FP='F: INPUT 'QP='Q
40 INPUT 'C2='C2: INPUT 'C3='C3: INPUT 'R6='R6
50 LET P1=1/(3*(1+C2/C3))
60 PRINT 'P НАЧ.ЗНАЧЕНИЕ ВБЛИЗИ MIN ПУЧ'P1
70 PRINT 'ВВЕДИТЕ КОД 0, ЕСЛИ P=PНАЧ'
80 INPUT 'ВВЕДИТЕ КОД 1, ЕСЛИ P ДРУГОЕ' A
90 IF A=0 THEN 120
100 IF A<>0 THEN 110
110 LET P=P1:GOTO 130
120 INPUT 'ВВЕДИТЕ P' P
130 LET R1=1/(2*(PI*F*SQR(C2*C3*P))):LETR=R1
140 IF R6=0 THEN LET R6=1E4
150 LET R7=R6*(1+(C2/C3))*P
160 LET B=1-((SQR(P*C2/C3))/Q*(1+R7/R6))
170 LET R5=R/B:LETR4=R/(1-B)
180 LET X=Q*B*((1+(R7/R6))^2)*SQR(C3/P/C2)
190 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА'
200 PRINT 'R1='R1:PRINT 'R4='R4:PRINT 'R5='R5
210 PRINT 'R7='R7:PRINT 'P='P:PRINT 'ПУЧ='X
220 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
230 PRINT 'T(S)=(S^2+W/P^2)/(S^2+(W/P/Q)*S+W/P^2)'
240 GOTO 120:END

```

$R_6=0$. Результаты расчета: $P=$
 $=69,566685 \cdot 10^{-3}$, ПУЧ=17,186565, $R_1=$
 $=27,33525 \cdot 10^3$, $R_4=13,255775 \cdot 10^3$, $R_5=$
 $=2,22201114 \cdot 10^3$, $R_7=4,3922515 \cdot 10^3$.

Если задано $R_6=0$, программа присваивает R_6 значение $1 \cdot 10^4$ Ом.

Пример А. Вводимые параметры:
 $F=1.9 \cdot 10^3$, $Q=9$, $C_2=68 \cdot 10^{-9}$, $C_3=15 \cdot 10^{-9}$,
 $R_6=5.6 \cdot 10^3$. Результаты расчета: $P=$
 $=60,240964 \cdot 10^{-3}$. ПУЧ=29,283848, $R_1=$
 $=10,686131 \cdot 10^3$, $R_4=14,782193 \cdot 10^3$, $R_7=$
 $=1,8666667 \cdot 10^3$, $R_5=673,05331$.

Пример В. Вводимые параметры:
 $F=1,9 \cdot 10^3$, $Q=9$, $C_2=15 \cdot 10^{-9}$, $C_3=$
 $=4,7 \cdot 10^{-9}$, $R_6=0$. Результаты расчета:
 $P=79,526227 \cdot 10^{-3}$, ПУЧ=30,42576, $R_1=$
 $=35,376656 \cdot 10^3$, $R_4=67,012608 \cdot 10^3$, $R_5=$
 $=2,936661 \cdot 10^3$, $R_7=3,333333 \cdot 10^3$.

Если задано $R_6=0$, программа присваивает R_6 значение $1 \cdot 10^4$ Ом. Настройка: (1) f_p резистором R_1 , (2) $|T(f_p)|=0$ резистором R_4 , (3) q_p резистором R_7 .

14. Низкочастотный и высокочастотный с нулем передачи.

$$T(s) = K(s^2 + \omega_2^2) / [s^2 + (\omega_p/q_p)s + \omega_p^2]$$

$$K = \frac{R_8}{R_7 + R_8}$$

$$\omega_p = \sqrt{\frac{R_1 + R_2}{R_1 R_2 C_3 C_4} \left[\frac{1}{R_5} - \frac{R_7 R_8}{R_6 R_9 (R_7 + R_8)} \right]}$$

$$\omega_z = \sqrt{\frac{(R_1 + R_2)(R_5 + R_6)}{R_1 R_2 C_3 C_4 R_8 R_6}} \quad \text{для НЧ.}$$

$$\omega_z = \sqrt{\frac{R_1 + R_2}{R_1 R_2 C_3 C_4} \left[\frac{1}{R_5} - \frac{R_7 (R_8 + R_9)}{R_6 R_8 R_9} \right]}$$

для ВЧ,

$$\frac{C_4}{R_2} + \frac{(C_3 + C_4)(R_5 + R_6)}{R_5 R_6} = \frac{C_4 R_7 (R_8 + R_9)}{R_1 R_8 R_9}$$

$$\frac{C_4}{R_2} + \frac{C_3 + C_4}{R_5} = \frac{R_7 (R_8 + R_9)}{R_8 R_9} \left(\frac{C_4}{R_1} + \frac{C_3 + C_4}{R_6} \right)$$

$$q_p = \omega_p / \left\{ \frac{C_3 + C_4}{C_3 C_4} \left[\frac{1}{R_5} - \frac{R_7 R_8}{R_6 R_9 (R_7 + R_8)} \right] - \frac{R_7 R_8 (R_1 + R_2)}{C_3 R_1 R_2 R_9 (R_7 + R_8)} \right\}$$

Программа 7.47.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА НЧ И ВЧ С 0 ПЕРЕДАЧИ СРЕДНЕДОБРОТНОГО'
03 PRINT'          !! C3
04 PRINT'          -----
05 PRINT'          !! 1
06 PRINT'          !! -
07 PRINT'          !! R5
08 PRINT'          !! C4 -
09 PRINT'          !! C4 -          ОУ
10 PRINT'          R1          !!
11 PRINT'          --!====!-*---!-*-----!
12 PRINT'          !!          !!          !!          !!          !!
13 PRINT'          !!          !!          !!          !!          !!
14 PRINT' <--*          R2          !!          !! R6          !!
15 PRINT'          !!          !!          !!          !!          !!
16 PRINT'          !!          !!          !!          !!          !!
17 PRINT'          !!          !!          !!          !!          R9
18 PRINT'          !!          !!          !!          !!          !!
19 PRINT'          !!          !!          !!          !!          !!
20 PRINT'          *-----* *          !!
21 PRINT'          !!          !!          !!          !!          !!
22 PRINT'          !!          R7          !!          R8
23 PRINT'          !-!====!-*-----!
30 INPUT'FZ='F0: INPUT'FP='F1: INPUT'QP='Q
40 INPUT'C3='C3: INPUT'C4='C4: INPUT'KBX='K1
45 INPUT'R9='R9
50 PRINT'ВЫБЕРИТЕ P В ИНТЕРВАЛЕ ОТ 0,1 ДО 0,3'
55 INPUT'P='P
60 IF(F1-F0)=0 THEN 80
70 IF(F1-F0)<>0 THEN 90
80 PRINT'ПОЛОСА ЗАДЕРЖИВАНИЯ ОТСУТСТВУЕТ':GOTO 30
90 LETX=(1+SGN(F1-F0))/2
100 LETB=2*#PI*F1:LETN=2*P*Q
110 LETG=(C3*D)/N*(SGR(1+4*(Q^2)*P*(1+C4/C3))-1)
120 LETM=2*#PI*F0
130 LETK0=(1+P)/(1+(1+C4/C3)*(M^2)*((C3^2)/(G^2)))
140 IF K1=0 THEN 170
150 LETH=1-(K1/K0)
160 IF H>0 THEN 180
170 LETK=K0:LETR1=1/6:LETR2=1E9:GOTO 190
180 LETK=K1:LETR1=K0/G/K1:LETR2=1/6/H
190 LETR6=(6*(1+P)*(1-X/K))/(C3*C4*(M^2-D^2))
200 LETR5=1/(C3*C4*(D^2)/6+P/R6)
210 LETH1=Q/D*(6/C3+(C3+C4)*(R5+R6)/C3/C4/R5/R6)
220 LETH2=1-6*(R5+R6)/(D^2)/C3/C4/R5/R6

```

```

230 LETJ=(1+P)/2*(ABS(1-H1)+Q*ABS(H2))
240 IF R9=0 THEN LETR9=1E4
250 LETR7=P*R9/K:LETR8=P*R9/(1-K)
260 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА'
270 PRINT'R1='R1:PRINT'R2='R2:PRINT'R5='R5
280 PRINT'R6='R6:PRINT'R7='R7:PRINT'R8='R8
290 PRINT'K='K:PRINT'P='P:PRINT'ПУЧ='J
300 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
310 PRINT'T(S)=K*(S^2+WZ^2)/(S^2+(WP/Q)*S+WP^2)'
320 END

```

Пример А (ВЧ-Д). Вводимые параметры: $F_z = 2.2 \cdot 10^3$, $F_p = 2.5 \cdot 10^3$, $Q = 10$, $P = 300 \cdot 10^{-3}$, $C_3 = 22 \cdot 10^{-9}$, $C_4 = 6.8 \cdot 10^{-9}$, $R_7 = 5 \cdot 10^3$, $R_8 = 7.5 \cdot 10^3$, $R_9 = 0$, $K_{\text{нз}} = 600 \cdot 10^{-3}$.
 Результаты расчета: ПУЧ = 19,991639, $R_1 = 2.554421 \cdot 10^3$, $R_2 = 3.6350018 \cdot 10^3$, $R_5 = 16.750422 \cdot 10^3$, $R_6 = 69.373675 \cdot 10^3$.

Пример В. Вводимые параметры: $F_z = 2.2 \cdot 10^3$, $F_p = 2.5 \cdot 10^3$, $Q = 10$, $P = 250 \cdot 10^{-3}$, $C_3 = 22 \cdot 10^{-9}$, $C_4 = 10 \cdot 10^{-9}$, $R_7 = 5 \cdot 10^3$, $R_8 = 5 \cdot 10^3$, $R_9 = 0$, $K_{\text{нз}} = 500 \cdot 10^{-3}$.
 Результаты расчета: ПУЧ = 19,680306, $R_1 = 2.6521947 \cdot 10^3$, $R_2 = 2.562408 \cdot 10^3$, $R_5 = 13.525052 \cdot 10^3$, $R_6 = 78.320603 \cdot 10^3$.

Пример С. Вводимые параметры: $F_z = 2.5 \cdot 10^3$, $F_p = 2.2 \cdot 10^3$, $Q = 6$, $P = 200 \cdot 10^{-3}$, $C_3 = 5.1 \cdot 10^{-9}$, $C_4 = 68 \cdot 10^{-9}$, $R_7 = 3 \cdot 10^3$, $R_8 = 3 \cdot 10^3$, $R_9 = 7.5 \cdot 10^3$, $K_{\text{нз}} = 500 \cdot 10^{-3}$.
 Результаты расчета: ПУЧ = 37,729721, $R_1 = 3.2874103 \cdot 10^3$, $R_2 = 3.7884357 \cdot 10^3$, $R_5 = 8.1769184 \cdot 10^3$, $R_6 = 35.317329 \cdot 10^3$.

Пример D (НЧ-Д). Вводимые параметры: $F_z = 2.5 \cdot 10^3$, $F_p = 2.2 \cdot 10^3$, $Q = 6$, $P = 200 \cdot 10^{-3}$, $C_3 = 18 \cdot 10^{-9}$, $C_4 = 10 \cdot 10^{-9}$, $R_7 = 4 \cdot 10^3$, $R_8 = 4 \cdot 10^3$, $R_9 = 0$, $K_{\text{нз}} = 0.5$.
 Результаты расчета: ПУЧ = 12,109246, $R_1 = 2.9779819 \cdot 10^3$, $R_2 = 3.8147434 \cdot 10^3$, $R_5 = 16.580147 \cdot 10^3$, $R_6 = 71.612124 \cdot 10^3$.

Если задано $R_9 = 0$, программа присваивает R_9 значение $1 \cdot 10^4$ Ом.

15. Полосно-заграждающие низкочастотный и высокочастотный с нулем передачи.
 $T(s) = K \frac{(s^2 + \omega_z^2)}{[s^2 + (\omega_p/q_p)s + \omega_p^2]}$,

$$C_s = C_1 C_2 / (C_1 + C_2),$$

$$K = (1 + R_{10}/R_9) / (1 + C_4/C_s),$$

$$R_s = R_5 + R_6,$$

$$\omega_z = 1/\sqrt{R_5 R_6 C_s C_3} = 1/\sqrt{R_5 R_7 C_1 C_2},$$

$$\omega_p = \omega_z \sqrt{\frac{1 + R_s/R_8}{1 + C_4/C_s}},$$

$$q^* = (2\sqrt{(1 + C_2/C_1)(1 + C_2/C_3)})^{-1},$$

$$q_p = q^* \frac{(1 + C_4/C_s)(\omega_p/\omega_z)}{q^*(1/R_8 C_s \omega_z + R_s C_4 \omega_z^2) - R_{10}/R_9},$$

ПУЧ =

$$= q_p \left(1 + \frac{R_{10}}{R_9}\right)^2 \frac{\sqrt{R_5 C_3 / R_6 C_s} + \sqrt{R_s C_2 / R_7 C_1}}{(1 + C_4/C_s)(\omega_p/\omega_z)}$$

Программа 7.48.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА ПОЛ-ЗАГР НЧ/ВЧ С 0 ПЕРЕДАЧИ 0 СРЕДНЕЕ'
02 PRINT'          ---!====!-*-!====!-
03 PRINT'          !      R9      !      R10      !
04 PRINT'          ---          !          !
05 PRINT'          !          !          !          !
06 PRINT'          !          !          !          !
07 PRINT'          !          !          !          !
08 PRINT'          !          !          !          !
09 PRINT'<---* C1      ! C2      *---*---*+ !
10 PRINT'          !!      !!      !!      !!      !
11 PRINT'          !---!-*-!---!---!      !
12 PRINT'          !!      !!      !!      !!      !
13 PRINT'          -      !      C4 === !!      !
14 PRINT'          R7 !!      == C3      !      !
15 PRINT'          !      !      !      !      !
16 PRINT'          !      !      !      !      !
17 PRINT'          !      !      !      !      !
18 PRINT'          !      !      !      !      !
30 INPUT'FZ='F0:INPUT'FP='F1:INPUT'QP='Q
40 INPUT'C1='C1:INPUT'C2='C2:INPUT'C3='C3
50 INPUT'C4='C4:INPUT'R9='R9
60 LETG=1/(2*SQR((1+C2/C1)*(1+C2/C3)))
70 LETR5=1/(2**PI*F0*G*(C2+C3))
80 LETC=C1*C2/(C1+C2):LETH1=1+C4/C:LETH2=1+C2/C1
90 LETR6=H2/R5/((2**PI*F0)^2)/C2/C3:LETB=R5+R6
100 LETR7=1/((2**PI*F0)^2)/C1/C2/B
105 LEH=H1*((2**PI*F1)^2)/((2**PI*F0)^2)-1
110 IF H<0 THEN PRINT'C4'=((F0^2)/(F1^2)-1)*C':GOTO 30
120 IF H=0 THEN LETR8=1E90:GOTO 140
130 LETR8=B/H

```

```

140 IF R9=0 THEN LETR9=1E4
145 LETN=2*#PI*F0:LETM=2*#PI*F1
150 LETR0=R9*G*(1/(R8*C*N)+B*C4*N-(H1*M/N)/Q)
160 LETB1=(2*#PI*F1)/(2*#PI*F0)
170 LETX=G*((1+R0/R9)^2)*(SQR(R5*C3/R6/C)+SQR(B*C2/R7/C1))/H1/B1
180 LETK=(1+R0/R9)/H1
190 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА'
200 PRINT'R5='R5:PRINT'R6='R6:PRINT'R7='R7:PRINT'R8='R8
210 PRINT'R10='R0:PRINT'K='K:PRINT'ПУЧ='X
220 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
230 PRINT'T(S)=K*(S^2+W^2)/(S^2+(W/Q)*S+W^2)'
240 END

```

Пример А (ВЧ-Д). Вводимые параметры: $F_z = 1,9 \cdot 10^3$, $F_p = 2,3 \cdot 10^3$, $Q = 5$, $C_1 =$

Программа 7.49.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА НЧ ВЫСОКОДОБРОТНОГО'
02 PRINT'
03 PRINT' ! C1 !
04 PRINT' ! !! !
05 PRINT' *---!-----*----- !
06 PRINT' ! !! ! ОУ ! ОУ !
07 PRINT' ! R1 ! R3 !
08 PRINT' *!====!-#-!- !-!====!-#-!- !
09 PRINT' ! ! ! !-# ! !-#->'
10 PRINT' !-!====!-#-!+ ! ! ! !-!+ !
11 PRINT' R2 ! ! !-! !-!-#-! !-! !
12 PRINT' ! ! ! ! !
13 PRINT' R6 ! ! C4 !
14 PRINT' ! ! ! R7 ! !
15 PRINT' ! ! ! ! !
16 PRINT' ! ! ! ! !
17 PRINT' ! ! ! ! !
18 PRINT' <-----!
30 INPUT'FF='F:INPUT'QP='Q:INPUT'C='C
40 LETR0=1/(2*#PI*F*C)
50 PRINT'R0='R0
55 PRINT'ВВЕДИТЕ R8 НОМИНАЛЬНОЕ ЗНАЧЕНИЕ R0'
60 INPUT'R8='R8
70 LETC1=C:LETC4=C:LETR2=R8:LETR3=R8:LETR6=R8
80 LETR1=Q*R0:LETR7=(R0^2)/R8
90 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА'
100 PRINT'R1='R1:PRINT'C1='C1:PRINT'R2='R2
110 PRINT'R3='R3:PRINT'C4='C4:PRINT'R6='R6
120 PRINT'R7='R7
130 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
140 PRINT'T(S)=K*(W^2)/(S^2+(W/Q)*S+W^2)'
150 END

```

$= 68 \cdot 10^{-9}$, $C_2 = 15 \cdot 10^{-9}$, $C_3 = 33 \cdot 10^{-9}$, $C_4 = 5,1 \cdot 10^{-9}$, $R_9 = 6,8 \cdot 10^3$. Результаты расчета: $R_5 = 2,3252708 \cdot 10^3$, $R_6 = 7,4408665 \cdot 10^3$, $R_7 = 704,38495$, $R_8 = 9,0974377 \cdot 10^3$, $R_{10} = 2,5549489 \cdot 10^3$, $K = 972,24578 \cdot 10^{-3}$, ПУЧ = 14,722532.

Пример В (НЧ-Д). Вводимые параметры: $F_z = 2,5 \cdot 10^3$, $F_p = 2,3 \cdot 10^3$, $Q = 12$, $C_1 = 18 \cdot 10^{-9}$, $C_2 = 4,7 \cdot 10^{-9}$, $C_3 = 18 \cdot 10^{-9}$, $C_4 = 4,7 \cdot 10^{-9}$, $R_9 = 0$. Результаты расчета: $R_5 = 3,5367765 \cdot 10^3$, $R_6 = 17,081878 \cdot 10^3$, $R_7 = 2,3234298 \cdot 10^3$, $R_8 = 22,563531 \cdot 10^3$, $R_{10} = 8,3494916 \cdot 10^3$, $K = 811,52543 \cdot 10^{-3}$, ПУЧ = 48,989492.

Если значение $R_9 = 0$, программа присваивает R_9 значение $1 \cdot 10^4$ Ом. Настройка: (1) f_z резистором R_5 , R_6 и R_7 (итеративно), (2) f_p резистором R_8 , (3) q_p резистором R_{10} .

Высокодооборотные фильтры.

16. Низкочастотный.

$T(s) = K \omega_p^2 / [s^2 + (\omega_p/q_p)s + \omega_p^2]$, $K = 1 + (R_2/R_6)$, $\omega_p = \sqrt{R_6/R_2 R_3 R_7 C_1 C_4}$, $q_p = \omega_p R_1 C_1$.

Пример А. Вводимые параметры: $F = 2,5 \cdot 10^3$, $Q = 30$, $C_1 = 68 \cdot 10^{-9}$, $R_2 = 1 \cdot 10^3$. Результаты расчета: $R_1 = 28,086166 \cdot 10^3$, $R_3 = 1 \cdot 10^3$, $C_4 = 68 \cdot 10^{-9}$, $R_5 = 1 \cdot 10^3$, $R_6 = 876,48083$.

Пример В. Вводимые параметры: $F = 2,5 \cdot 10^3$, $Q = 30$, $C_1 = 2,2 \cdot 10^{-9}$, $R_2 = 2,7 \cdot 10^3$. Результаты расчета: $R_1 = 86,811787$, $R_3 = 2,7 \cdot 10^3$, $C_4 = 22 \cdot 10^{-9}$, $R_6 = 2,7 \cdot 10^3$, $R_7 = 3,1013524 \cdot 10^3$.

Настройка: (1) f_p резистором R_7 , (2) q_p резистором R_1 .

Примечание. Для фильтров 16—22 R_0 представляет собой оптимальное значение резистора для низкого значения ПУЧ, а R_d является его номинальным значением, которое выбирается как можно ближе к R_0 .

17. Полосно-пропускающий.

$T(s) = K (\omega_p/q_p) s / [s^2 + (\omega_p/q_p)s + \omega_p^2]$, $K = 1 + R_2/R_6$, $\omega_p = \sqrt{R_2/R_1 R_4 R_6 C_3 C_8}$, $q_p = \omega_p R_7 C_8$.

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ПОЛОСНО-ПРОПУСКАЮЩЕГО'
02 PRINT 'ВЫСОКОЧЕБОРНОГО'
03 PRINT '-----'
04 PRINT ' |           |           |           |           |'
05 PRINT ' |           |           |           |           |'
06 PRINT ' |           |           |           |           |'
07 PRINT ' |           |           |           |           |'
08 PRINT ' |           |           |           |           |'
09 PRINT ' |           |           |           |           |'
10 PRINT ' |           |           |           |           |'
11 PRINT ' |           |           |           |           |'
12 PRINT ' |           |           |           |           |'
13 PRINT ' |           |           |           |           |'
14 PRINT ' |           |           |           |           |'
15 PRINT ' |           |           |           |           |'
16 PRINT ' |           |           |           |           |'
17 PRINT ' <-----'
30 INPUT 'FP='F: INPUT 'QP='Q: INPUT 'C='C
40 LETR0=1/(2*PI*F*C)
50 PRINT 'R0='R0
60 PRINT 'ВВЕДИТЕ R8 НОМИНАЛЬНОЕ ЗНАЧЕНИЕ R0'
70 INPUT 'R8='R8
80 LETC3=C: LETC8=C: LETR1=R8: LETR2=R8: LETR6=R8
90 LETR7=Q*R0: LETR4=(R0^2)/R8
100 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА'
110 PRINT 'R1='R1: PRINT 'R2='R2: PRINT 'C3='C3
120 PRINT 'R4='R4: PRINT 'R6='R6: PRINT 'R7='R7
130 PRINT 'C8='C8
140 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
150 PRINT 'T(S)=K*(WP/Q)*S/(S^2+(WP/Q)*S+WP^2)'
160 END

```

Пример А. Вводимые параметры:
 $F=2.2 \cdot 10^3$, $Q=25$, $R_1=3.3 \cdot 10^3$, $C_3=22 \cdot 10^{-9}$.
 Результаты расчета: $R_2=3.3 \cdot 10^3$, $R_4=$
 $=3.2766918 \cdot 10^3$, $R_6=3.3 \cdot 10^3$, $R_7=$
 $=82.208132 \cdot 10^3$, $C_8=22 \cdot 10^{-9}$.

$=2,6589098 \cdot 10^3$, $R_6=2,7 \cdot 10^3$, $R_7=$
 $=160,76257 \cdot 10^3$, $C_8=2,7 \cdot 10^{-9}$.
 Настройка: (1) f_p резистором R_4 , (2) q_p
 резистором R_7 .

Пример В. Вводимые параметры:
 $F=2.2 \cdot 10^3$, $Q=60$, $R_1=2.7 \cdot 10^3$, $C_3=27 \cdot 10^{-9}$.
 Результаты расчета: $R_2=2,7 \cdot 10^3$, $R_4=$

18. *Высокочастотный.*
 $T(s) = Ks^2 / [s^2 + (\omega_p/q_p)s + \omega_p^2]$, $K=1+R_2/R_6$,
 $\omega_p = \sqrt{R_2/R_1 R_4 R_6 C_3 C_7}$, $q_p = \omega_p R_8 C_7$.
 Программа 7.51.

```

01 PRINT 'РАСЧЕТ ФИЛЬТРА ВЧ ВЫСОКОЧЕБОРНОГО'
02 PRINT '-----'
03 PRINT ' |           |           |           |           |'
04 PRINT ' |           |           |           |           |'
05 PRINT ' |           |           |           |           |'
06 PRINT ' |           |           |           |           |'
07 PRINT ' |           |           |           |           |'
08 PRINT ' |           |           |           |           |'
09 PRINT ' |           |           |           |           |'
10 PRINT ' |           |           |           |           |'
11 PRINT ' |           |           |           |           |'
12 PRINT ' |           |           |           |           |'
13 PRINT ' |           |           |           |           |'
14 PRINT ' |           |           |           |           |'
15 PRINT ' |           |           |           |           |'
16 PRINT ' <-----'
20 INPUT 'FP='F: INPUT 'QP='Q: INPUT 'C='C
30 LETR0=1/(2*PI*F*C)
40 PRINT 'R0='R0
50 PRINT 'ВВЕДИТЕ R9 НОМИНАЛЬНОЕ ЗНАЧЕНИЕ R0'
60 INPUT 'R9='R9
70 LETC3=C: LETC7=C: LETR1=R9: LETR2=R9: LETR6=R9
80 LETR8=Q*R0: LETR4=(R0^2)/R9
90 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА'
100 PRINT 'R1='R1: PRINT 'R2='R2: PRINT 'C3='C3
110 PRINT 'R4='R4: PRINT 'R6='R6: PRINT 'C7='C7
120 PRINT 'R8='R8
130 PRINT 'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
140 PRINT 'T(S)=K*(S^2)/(S^2+(WP/Q)*S+WP^2)'
150 END

```

Пример А. Вводимые параметры:
 $F=1,9 \cdot 10^3$, $Q=20$, $R_1=1,5 \cdot 10^3$, $C_3=68 \cdot 10^{-9}$.
 Результаты расчета: $R_2=1,5 \cdot 10^3$, $R_4=$
 $=1,0116353 \cdot 10^3$, $R_6=1,5 \cdot 10^3$, $C_7=68 \cdot 10^{-9}$,
 $R_8=24,636988 \cdot 10^3$.

Настройка: (1) f_p резистором R_4 , (2) q_p резистором R_6 .

19. *Всепропускающий.*

$$T(s) = [s^2 - (\omega_p/q_p)s + \omega_p^2] / [s^2 + (\omega_p/q_p)s + \omega_p^2]$$

$$\omega_p = \sqrt{R_2/R_1 R_4 R_5 C_3 C_7}, \quad q_p = \omega_p C_7 R_8, \quad R_2 = R_5.$$

Программа 7.52.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА ВСЕПРОПУСКАЮЩЕГО ВЫСОКОДОБОРНОГО'
02 PRINT'-----'
03 PRINT'!'
04 PRINT'!      !      ОУ      C3      !      ОУ      !'
05 PRINT'!      R1      !-----!      !!      !-----!      !'
06 PRINT'! *!====!-!*!-! !-!!-!-!*-----!-! !-! !-!->'
07 PRINT'!      !      !      !      !-!*      !!      !      !      !-!->'
08 PRINT'!-!====!-!*!+! ! ! !====!-!*-*! !-!+!'
09 PRINT'!      R2      !-----!      !-!*-----!*-*! !-----!'
10 PRINT'!      -      !      !      R4      !      -      !'
11 PRINT'!      R5      !!      !      !      !      !      !'
12 PRINT'!      !!      !      !      C7      ===      !!      R8'
13 PRINT'!      -      !      !      !      !      !      !'
14 PRINT'!      !      !      !      !      !      !      !'
15 PRINT'!      !      !      !      !      !      !      !'
16 PRINT'! <-----!*-----!*-----!'
20 INPUT'FP='F: INPUT'QP='Q: INPUT'C='C
30 LETR0=1/(2*PI*F*C)
40 PRINT'R0='R0
50 PRINT'ВВЕДИТЕ R9 НОМИНАЛЬНОЕ ЗНАЧЕНИЕ R0'
60 INPUT'R9='R9
70 LETC3=C: LETC7=C: LETR1=R9: LETR2=R9: LETR5=R9
80 LETR0=Q*R0: LETR4=(R0^2)/R9
90 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА'
100 PRINT'R1='R1: PRINT'R2='R2: PRINT'C3='C3
110 PRINT'R4='R4: PRINT'R5='R5: PRINT'C7='C7
120 PRINT'R8='R8
130 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
140 PRINT'T(S)=(S^2-(WP/Q)*S+WP^2)/(S^2+(WP/Q)*S+WP^2)'
150 END
    
```

Пример А. Вводимые параметры:
 $F=2 \cdot 10^3$, $Q=25$, $R_1=1,2 \cdot 10^3$, $C_3=68 \cdot 10^{-9}$.
 Результаты расчета: $R_2=1,2 \cdot 10^3$, $R_4=$

$=1,1412511 \cdot 10^3$, $R_6=1,2 \cdot 10^3$, $C_7=68 \cdot 10^{-9}$,
 $R_8=29,256423 \cdot 10^3$.

Пример В. Вводимые параметры:
 $F=2 \cdot 10^3$, $Q=25$, $R_1=4,7 \cdot 10^3$, $C_3=18 \cdot 10^{-9}$.
 Результаты расчета: $R_2=4,7 \cdot 10^3$, $R_4=$
 $=4,1585067 \cdot 10^3$, $R_6=4,7 \cdot 10^3$, $C_7=18 \cdot 10^{-9}$,
 $R_8=110,52427 \cdot 10^3$.

Настройка: (1) f_p резистором R_4 , (2) q_p резистором R_6 .

20. *Полосно-заграждающий.*

$$T(s) = (s^2 + \omega_p^2) / [s^2 + (\omega_p/q_p)s + \omega_p^2]$$

$$\omega_p = \sqrt{R_2/R_1 R_4 R_5 C_3 C_7},$$

$$q_p = \omega_p C_7 [R_7 R_8 / (R_7 + R_8)], \quad R_2 R_7 = R_5 R_8.$$

Программа 7.53.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА ПОЛОСНО-ЗАГРАЖДАЮЩЕГО'
02 PRINT'ВЫСОКОДОБОРНОГО'
03 PRINT'-----'
04 PRINT'!'
05 PRINT'!      !      ОУ      C3      !      ОУ      !'
06 PRINT'!      R1      !-----!      !!      !-----!      !'
07 PRINT'! *!====!-!*!-! !-!!-!-!*-----!-! !-! !-!->'
08 PRINT'!      !      !      !      !-!*      !!      !      !      !-!->'
09 PRINT'!-!====!-!*!+! ! ! !====!-!*-*! !-!+!'
10 PRINT'!      R2      !-----!      !-!*-----!*-*! !-----!'
11 PRINT'!      -      !      !      R4      !      -      !'
12 PRINT'!      R5      !!      !      !      !      !      !'
13 PRINT'!      !!      !      !      C7      ===      !!      R7      !!      R8'
14 PRINT'!      -      !      !      !      !      !      !'
15 PRINT'!      !      !      !      !      !      !      !'
16 PRINT'!      !      !      !      !      !      !      !'
17 PRINT'! <-----!*-----!*-----!'
30 INPUT'FP='F: INPUT'QP='Q: INPUT'C='C
40 LETR0=1/(2*PI*F*C)
50 PRINT'R0='R0
60 PRINT'ВВЕДИТЕ НОМИНАЛЬНОЕ ЗНАЧЕНИЕ R0'
    
```



```

70 INPUT R9='R9
80 LETC3=C3:LETC7=C7:LETR1=R9:LETR2=R9:LETR5=R9
90 LETR7=2*Q*R0:LETR8=2*Q*R0:LETR4=(R0^2)/R9
100 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА'
110 PRINT'R1='R1:PRINT'R2='R2:PRINT'C3='C3
120 PRINT'R4='R4:PRINT'R5='R5:PRINT'R7='R7
130 PRINT'C7='C7:PRINT'R8='R8
140 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
150 PRINT'T(S)=(S^2+WP^2)/(S^2+(WP/Q)*S+WP^2)'
160 END

```

Пример А. Вводимые параметры:
 $F=2,2 \cdot 10^3$, $Q=20$, $R_1=2,2 \cdot 10^3$, $C_3=33 \cdot 10^{-9}$.

Результаты расчета: $R_2=2,2 \cdot 10^3$,
 $R_4=2,1844612 \cdot 10^3$, $R_5=2,2 \cdot 10^3$,
 $R_7=87,688674 \cdot 10^3$, $C_7=33 \cdot 10^{-9}$,
 $R_8=87,688674 \cdot 10^3$.

21. Низкочастотный и высокочастотный с нулем передачи.

$$T(s) = \frac{(s^2 + \omega_0^2)}{[s^2 + (\omega_p/q_p)s + \omega_0^2]},$$

$$\omega_p = \sqrt{R_3/R_1 R_4 R_5 C_2 C_7},$$

$$q_p = \omega_p C_7 R_8, \quad \omega_{2нч} = \omega_p \sqrt{1 + R_4/R_8}, \quad \omega_{2вч} = \omega_p \sqrt{1 - R_1 R_4/R_3 R_8}.$$

Программа 7.54.

```

01 PRINT'РАСЧЕТ ФИЛЬТРА НЧ И ВЧ С НУЛЕМ ПЕРЕДАЧИ'
02 PRINT'ВЫСОКОДЕБЕРНОГО'
03 PRINT'
04 PRINT' ! ----- !
05 PRINT' !           ! ОУ           ! ----- !
06 PRINT' !           ! R1           ! ----- !
07 PRINT' * !===== ! * !----- ! *----- ! ----- !
08 PRINT' !           ! R3           ! ----- ! ----- ! *----- !
09 PRINT' !           ! +----- ! +----- ! ----- !
10 PRINT' !           ! R4           ! ----- ! ----- ! ОУ
11 PRINT'           !           ! R5 <- ! ----- !
12 PRINT'           !           ! R7 == ! ----- ! R8
13 PRINT'           !           ! ----- ! ----- !
14 PRINT'           !           ! ----- ! ----- !
15 PRINT'           !           ! ----- ! ----- !
16 PRINT'           !           ! ----- ! ----- !
17 PRINT' <----- *----- ! ----- !
30 INPUT'FZ='F0:INPUT'FP='F1:INPUT'QP='Q
40 INPUT'C='C
50 LETR0=1/(2*#PI*F1*C)
60 PRINT'R0='R0
70 PRINT'ВВЕДИТЕ НОМИНАЛЬНОЕ ЗНАЧЕНИЕ R0'
80 INPUT'R9='R9
90 LETR1=R9:LETR3=R9
100 LETC2=C:LETC7=C:LETR8=Q*R0
110 IF F0<>F1 THEN 130
120 PRINT'ПОЛОСА ЗАДЕРЖИВАНИЯ ОТСУТСТВУЕТ':GOTO 30
130 IF ((F0^2)/(F1^2)-1)>0 THEN 160
140 LETR4=R8*(1-((F0^2)/(F1^2)))
150 PRINT'ВЧ-Д':GOTO 180
160 LETR4=R8*((F0^2)/(F1^2)-1)
170 PRINT'НЧ-Д'
180 LETR5=(R0^2)/R4
190 PRINT'РЕЗУЛЬТАТЫ РАСЧЕТА'
200 PRINT'R1='R1:PRINT'C2='C2:PRINT'R3='R3
210 PRINT'R4='R4:PRINT'R5='R5:PRINT'C7='C7
220 PRINT'R8='R8
230 PRINT'ПЕРЕДАТОЧНАЯ ФУНКЦИЯ ИМЕЕТ ВИД'
240 PRINT'T(S)=(S^2+WZ^2)/(S^2+(WP/Q)*S+WP^2)'
250 END

```

Пример В. Вводимые параметры:
 $F=2,2 \cdot 10^3$, $Q=20$, $R_1=15 \cdot 10^3$, $C_3=4,7 \cdot 10^{-9}$.

Результаты расчета: $R_2=15 \cdot 10^3$,
 $R_4=15,794574 \cdot 10^3$, $R_5=15 \cdot 10^3$,
 $R_7=615,68643 \cdot 10^3$, $C_7=4,7 \cdot 10^{-9}$,
 $R_8=615,68643 \cdot 10^3$.

Настройка: (1) f_p резистором R_4 , (2) $[T(f_p)] = 0$ и q_p резистором R_7 и R_8 (итеративно).

Пример А. Вводимые параметры:
 $F_2=2,2 \cdot 10^3$, $F_p=2,5 \cdot 10^3$, $Q=15$, $R_1=2,7 \cdot 10^3$,
 $C_2=22 \cdot 10^{-9}$. Результаты расчета: $R_3=$
 $=2,7 \cdot 10^3$, $R_4=9,7923696 \cdot 10^3$, $R_5=855,12005$,
 $C_7=22 \cdot 10^{-9}$, $R_8=43,405894 \cdot 10^3$.

Пример В (НЧ-Д). Вводимые параметры: $F_2=2,5 \cdot 10^3$, $F_p=2,2 \cdot 10^3$, $Q=15$,
 $R_1=3,3 \cdot 10^3$, $C_2=22 \cdot 10^{-9}$. Результаты расчета: $R_3=3,3 \cdot 10^3$, $R_4=14,369438 \cdot 10^3$, $R_5=$
 $=752,50564$, $C_7=22 \cdot 10^{-9}$, $R_8=49,324879 \cdot 10^3$.


```

01 PRINT ' РАСЧЕТ ФИЛЬТРА УНИВЕРСАЛЬНОГО'
02 PRINT ' СХЕМА 2'
03 PRINT'
04 PRINT'
05 PRINT'
06 PRINT'
07 PRINT'
08 PRINT'
09 PRINT'
10 PRINT'
11 PRINT'
12 PRINT'
13 PRINT'
14 PRINT'
15 PRINT'
16 PRINT'
17 PRINT'
18 PRINT'
19 PRINT'
20 PRINT'
21 PRINT'
22 PRINT'
23 PRINT'
24 PRINT'
25 PRINT'
26 PRINT'
27 PRINT'
28 PRINT'
29 PRINT'
30 INPUT 'FP=' F: INPUT 'QF=' Q: INPUT 'C=' C: INPUT 'K=' K
40 LET R0=1/(2*PI*F*C)
50 PRINT 'R0=' R0
55 PRINT 'ВВЕДИТЕ НОМИНАЛЬНОЕ ЗНАЧЕНИЕ R0'
60 INPUT 'R9=' R9
70 LET R2=R9: LET R7=R9: LET R8=R9
80 LET C3=C: LET C6=C
90 LET R5=(R0^2)/R9: LET R4=Q*R0
100 PRINT 'ВАРИАНТ А'
110 LET R1=R4/K: LET K1=K
120 LET K2=R2/R1: LET K3=R2/R1
121 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА'
122 PRINT 'R1=' R1: PRINT 'R2=' R2: PRINT 'C3=' C3
123 PRINT 'R4=' R4: PRINT 'R5=' R5: PRINT 'C6=' C6
124 PRINT 'R7=' R7: PRINT 'R8=' R8: PRINT 'КПП=' K1
125 PRINT 'КНЧ1=' K2: PRINT 'КНЧ2=' K3: INPUT 'X'
126 PRINT 'ВАРИАНТ В'
130 LET R1=R2/K: LET K1=R4/R1
140 LET K2=K: LET K3=K
150 PRINT 'РЕЗУЛЬТАТЫ РАСЧЕТА'
160 PRINT 'R1=' R1: PRINT 'R2=' R2: PRINT 'C3=' C3
170 PRINT 'R4=' R4: PRINT 'R5=' R5: PRINT 'C6=' C6
180 PRINT 'R7=' R7: PRINT 'R8=' R8: PRINT 'КПП=' K1
190 PRINT 'КНЧ1=' K2: PRINT 'КНЧ2=' K3
200 PRINT 'ПЕРЕДАТОЧНЫЕ ФУНКЦИИ ИМЕЮТ ВИД'
210 PRINT 'ТПП(S)=-КПП*(WP/Q)*S/(S^2+(WP/Q)*S+WP^2)'
220 PRINT 'ТНЧ1(S)=КНЧ1*(WP^2/(S^2+(WP/Q)*S+WP^2))'
230 PRINT 'ТНЧ2(S)=-КНЧ2*(WP^2/(S^2+(WP/Q)*S+WP^2))'

```

Пример А. Вводимые параметры: $F=2,5 \cdot 10^3$, $Q=9$, $R_1=1 \cdot 10^3$, $R_2=1,5 \cdot 10^3$, $C_3=47 \cdot 10^{-9}$. Результаты расчета: $R_4=12,190591 \cdot 10^3$, $R_5=1,2231318 \cdot 10^3$, $C_6=47 \cdot 10^{-9}$, $R_7=1,5 \cdot 10^3$, $R_8=1,5 \cdot 10^3$, КПП= $=12,190591$, КНЧ= $1,5$.

Пример В. Вводимые параметры: $F=2,5 \cdot 10^3$, $Q=9$, $R_2=1,5 \cdot 10^3$, $C_3=47 \cdot 10^{-9}$. Результаты расчета: $R_1=8,1270609 \cdot 10^3$, $R_4=$

$=12,190591 \cdot 10^3$, $R_5=1,2231318 \cdot 10^3$, $C_6=47 \cdot 10^{-9}$, $R_7=1,5 \cdot 10^3$, $R_8=1,5 \cdot 10^3$, КПП= $=1,5$, КНЧ= $184,56857 \cdot 10^{-3}$.

Настройка: (1) f_p резистором R_5 , (2) q_p резистором R_4 , (3) K резистором R_1 .

Если нет необходимости в выдаче на экран дисплея (или в распечатке принтером) принципиальных схем фильтров, программы могут быть существенно сокращены.

§ 7.7 Расчет нелинейных и ключевых электронных устройств

В отличие от расчета линейных электронных устройств, базирующегося в основном на использовании аналитических методов, расчет нелинейных устройств обычно требует применения различных численных методов.

Например, семейство выходных вольт-амперных характеристик мощного МДП-транзистора с горизонтальным каналом хорошо аппроксимируется выражением [10]

$$I_c(U_{сн}, U_{зн}) = S_0(U_{зн} - bU_{зн}^2) \{1 - e^{-pU_{сн}/(U_{зн} - bU_{зн}^2)}\},$$

где p и b — параметры аппроксимации. Однако из этого выражения нельзя найти напряжение на затворе $U_{зн}$, при котором $I_c(U_{сн0}, U_{зн0}) = I_{c0}$ и $U_{сн} = U_{сн0}$. Эта задача решается численным методом решения нелинейного уравнения $I_c - I_{c0}(U_{сн0}, U_{зн}) = 0$ или, после простых преобразований, находится из уравнения

$$U_{зн} = \frac{-I_{c0}}{S_0 \{e^{-pU_{сн0}/(U_{зн} - bU_{зн}^2)} - 1\}} + bU_{зн}^2.$$

В таком виде уравнение может решаться методом простых итераций. Процедура расчета $U_{зн0}$ целесообразно объединить с нахождением малосигнальной крутизны $S = \partial I_c / \partial U_{зн}$ и выходного сопротивления $R_i = \partial U_{сн} / \partial I_c$. Для этого току I_{c0} придается малое приращение ΔI_c и находится новое значение $U'_{зн0}$. Тогда $S \approx \Delta I_c / (U'_{зн0} - U_{зн0})$. Далее, задав $U_{сн0}$ малое приращение $\Delta U_{сн}$, из формулы для $I_c(U_{сн}, U_{зн0})$ находим приращение тока стока ΔI_c и $R_i \approx \Delta U_{сн} / \Delta I_c$.

Программа 7.57.

```

10 PRINT'РАСЧЕТ ПАРАМЕТРОВ МОЩНОГО МДП-ТРАНЗИСТОРА'
20 INPUT'ВВЕДИТЕ IC0,UC0,S0,I0,U0,S0
30 INPUT'ВВЕДИТЕ ПРИРАЩЕНИЯ IC,UC'X,Y
40 INPUT'ВВЕДИТЕ ПОСТОЯННЫЕ P,B'P,B
50 INPUT'ВВЕДИТЕ ТОЧНОСТЬ РАСЧЕТА H'H
60 LET I1=I0:LET U1=U0:S0:GOSUB 110
65 LET A2=A:LET A3=A1
70 PRINT'UZHO=U2:LET U3=U2:LET I1=I0+X:GOSUB 110
80 LET S1=X/(U2-U3):PRINT'S=S1
90 LET F=(U0+Y)*P/A2:LET X1=S0*A2*(EXP(F)-EXP(A3))
100 LET R=Y/X1:PRINT'R=R:END
115 LET A=B*U1^2-U1:LET A1=P*U0/A
120 LET E=S0*(1-EXP(A1)):LET U2=I1/E+B*U1^2
130 LET K=(U1-U2)^2-H^2:LET U1=U2
140 IF K>0 THEN 115
150 RETURN:END
    
```

Для транзистора КП905, имеющего $S_0 = 0,03$ А/В, $p = 1$ и $b = 0,02$ 1/В при $I_{c0} = 0,1$ А, $U_{сн0} = 10$ В, $\Delta U_{сн} = 1$ В и $\Delta I_c = 0,01$ А при погрешности $H = 0,001$, находим $U_{зн0} = 3,839$ В, $S = 18,625 \cdot 10^{-3}$ А/В и $R_i = 643,3$ Ом.

Более сложной является задача моделирования работы нелинейных усилителей при усилении ими синусоидального сигнала, например, трансформаторного однотактного кас-

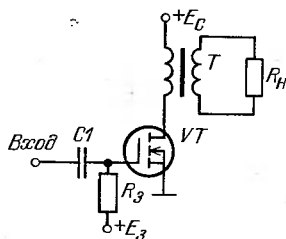


Рис. 7.16. Однотактный трансформаторный каскад усиления мощности на мощном U -МДП-транзисторе

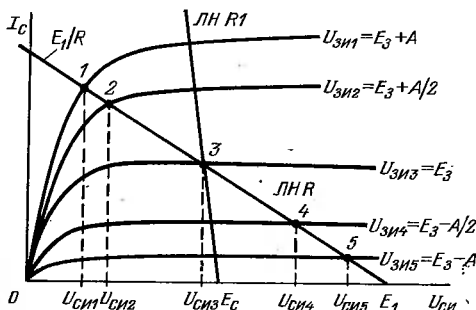


Рис. 7.17. Графические построения к расчету трансформаторного каскада

када на мощном U -МДП-транзисторе (рис. 7.16). Семейство выходных вольт-амперных характеристик последнего (рис. 7.17)

аппроксимируется выражением

$$I_c(U_{сн}, U_{зн}) = M(U_{зн}) (1 - e^{-kU_{сн}S/M(U_{зн})}), \quad (7.18)$$

$$M(U_{зн}) = I_s \left[1 + \text{th} \frac{S(U_{зн} - U_s)}{I_s} \right], \quad (7.19)$$

где U_s и I_s — координаты точки перегиба передаточной характеристики (рис. 7.18), K — коэффициент аппроксимации.

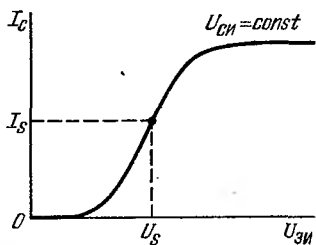


Рис. 7.18. Передаточная характеристика мощного U-МДП-транзистора

Алгоритм моделирования каскада, изображенного на рис. 7.16.

1. Вводим исходные данные: параметры транзистора I_s , U_s , S и $D=K$, сопротивления первичной R_1 и вторичной R_2 обмоток трансформатора, коэффициент трансформации N , сопротивление нагрузки R_n , напряжение питания стока E_c , напряжение смещения затвора E_3 , амплитуду входного синусоидального сигнала A .

2. Задав $U_{зи}=E_3$, $R=R_1$, $E_1=E_c$ и обращаясь к подпрограмме решения нелинейного уравнения

$$F(U_{си}, U_{зи}) = E_1 - U_{си} - R_n I_c(U_{зи}, U_{си}) = 0, \quad (7.20)$$

находим напряжение $U_{си0}$ и ток $I_{с0}=I_{с0}$ (см. рис. 7.17) в точке покоя.

3. Находим приведенное в первичную цепь трансформатора сопротивление динамической нагрузки (прямая B на рис. 7.17) $R=R_1+(R_2+R_n)/N^2$ и напряжение $E_1=E_c+RI_{с0}$.

5. Находим остальные четыре точки пересечения прямой динамической нагрузки и лий выходных вольт-амперных характеристик. Для этого задаем четыре значения мгновенного напряжения на затворе: $U_{зи1}=E_3+A$, $U_{зи2}=E_3+A/2$, $U_{зи4}=E_3-A/2$ и $U_{зи5}=-E_3-A$. Для этих значений $U_{зи}$, обращаясь к подпрограмме решения уравнения (7.20), находим значения $U_{си1}$, $U_{си2}$, $U_{си4}$ и $U_{си5}$.

5. Методом пяти ординат находим амплитуды первых четырех гармоник спектра переменного напряжения на стоке, коэффициент нелинейных искажений

$$K_T = \frac{\sqrt{U_{см2}^2 + U_{см3}^2 + U_{см4}^2}}{U_{см1}}$$

выходную мощность (полную) $P_2=PN=$
 $= (U_{си1}^2 + U_{си2}^2 + U_{си3}^2 + U_{си4}^2)/2R$,

мощность в нагрузке $P_n=P_2P_u/(RN^2)$, потребляемую каскадом мощность $P_o=$
 $=I_{с0}E_c$,

рассеиваемую транзистором мощность $P_T=P_o-P_2$,

коэффициент полезного действия к. п. д. = $=P_2/P_o$,

коэффициент усиления каскада по первой гармонике $K_U=U_{си1}/A=U_{си1}R_n/(ANR)$.

Проводя расчеты для разных A , можно построить зависимости основных параметров каскада (K_T , P_n , P_o , к. п. д. и др.) от амплитуды напряжения возбуждения. Решение (7.20) в приведенной ниже программе выполняется комбинированным методом секущих — хорд при погрешности $H=1 \cdot 10^{-4}$.

Программа 7.58.

```

10 PRINT 'МОДЕЛИРОВАНИЕ ТРАНСФОРМАТОРНОГО ОДНОТАКТНОГО'
20 PRINT 'УСИЛИТЕЛЯ МОЩНОСТИ НА U-МДП-ТРАНЗИСТОРЕ'
30 INPUT 'ВВЕДИТЕ ПАРАМЕТРЫ ТРАНЗИСТОРА I_s, U_s, S, I_1, U_1, S'
40 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЯ ОБМОТОК ТРАНСФОРМАТОРА R_1, R_2' R_1, R_2
50 INPUT 'ВВЕДИТЕ КОЭФФИЦИЕНТ ТРАНСФОРМАЦИИ N=' N
60 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ НАГРУЗКИ R_n=' R_3
70 INPUT 'ВВЕДИТЕ НАПРЯЖЕНИЕ ПИТАНИЯ СТОКА E_c=' E
80 INPUT 'ВВЕДИТЕ НАПРЯЖЕНИЕ СМЕЩЕНИЯ ЗАТВОРА E_3=' E_3
90 INPUT 'ВВЕДИТЕ АМПЛИТУДУ ВХОДНОГО СИГНАЛА A=' A
100 LET U=E3 : LET R=R1 : LET E1=E : GOSUB 290
110 LET I2=I : LET Y3=X1 : LET R=(R1+(R2+R3)/(N^2)): LET E1=X1+I*R
120 LET U=E3+A : GOSUB 290
130 LET Y1=X1 : LET U=E3+A/2 : GOSUB 290
140 LET Y2=X1 : LET U=E3-A/2 : GOSUB 290
150 LET Y4=X1 : LET U=E3-A : GOSUB 290
160 LET Y5=X1 : REM 'МЕТОД ПЯТИ ОРДИНАТ'
170 LETM0=(Y1+Y5+2*(Y2+Y4))/6: LETM1=(Y1-Y5+Y2-Y4)/3
180 LETM2=(Y1+Y5-2*Y3)/4: LETM3=(Y1-Y5-2*(Y2-Y4))/6
210 LET M4=(Y1+Y5)-4*(Y2+Y4)+6*Y3/12
220 LETK=(SQR((M2^2)+(M3^2)+(M4^2)))/ABS(M1)
230 LET P1=((M1^2)/(2*R))*(R3/R): LET P0=I2*E
240 PRINT 'КОЭФФИЦИЕНТ ГАРМОНИК K_T=' K: PRINT 'МОЩНОСТИ'
250 PRINT 'В НАГРУЗКЕ P_n=' P1: PRINT 'ПОТРЕБЛЯЕМАЯ P_o=' P0
260 PRINT 'РАССЕИВАЕМАЯ ТРАНЗИСТОРОМ P_T=' P0-P1
270 PRINT 'КОЭФФИЦИЕНТ ПОЛЕЗНОГО ДЕЙСТВИЯ К.П.Д.=' P1/P0
275 LETU=ABS(M1)*R3/A/N/R: PRINT 'КОЭФФИЦИЕНТ УСИЛЕНИЯ K_U=' U
280 PRINT: GOTO 90
285 REM 'КОМБИНИРОВАННЫЙ МЕТОД СЕКУЩИХ-ХОРД'
290 LET X0=0 : LET X1=1 : LET H=1E-4
300 LET X=X0 : GOSUB 350
310 LET C=F : LET X=X1 : GOSUB 350
320 LET B=F : LET Y=X0-C*(X1-X0)/(B-C)

```

```

330 LET X0=X1 : LET X1=Y : IF ABS(X1-X0)>HN THEN 300
340 RETURN
350 LET Z=(I1*(1+HTN(S*(U-U1)/I1)))
360 LET I=Z*(1-EXP((-0.7*S*X)/Z))
370 LET F=E1-X-I*R:RETURN:END

```

Пример. Для $I_s=2$ А, $U_s=10$ В, $S=1$ А/В, $D=0,7$, $R_1=R_2=0,1$ Ом, $N=2$, $R_n=15$ Ом, $E_c=40$ В, $E_s=10$ В и $A=3$ В получим $K_r=0,1192$, $R_n=31,521$ Вт, $P_o=80$ Вт, $P_r=48,48$ Вт, к. п. д. $=0,394$ и $K_U=5,125$.

Нелинейный режим работы m -фазного выпрямителя (см., например, рис. 7.19) ($m=2$) обусловлен как резкой нелинейностью диодов, так и отпиранием их в течение небольшой части периода входного синусоидального напряжения. Расчет выпрямителя проводится по следующему алгоритму.

1. Вводим исходные данные: сопротивление нагрузки R_n , внутреннее сопротивление (диодов и обмоток трансформатора, пересчитанных в цепь вторичной обмотки) r , частоту

отсечки θ , характеризующий ту часть периода, в течение которой отпираются диоды.

3. Вычисляются коэффициенты: $K(\theta)=\sin \theta-\theta \cos \theta$, $B(\theta)=[(\cos \theta) \sqrt{2}]^{-1}$,

$$F(\theta)=\pi(1-\cos \theta) / K(\theta),$$

$$D(\theta)=\frac{\sqrt{\pi}[\theta(1+(\cos 2\theta) / 2)-(3 \sin 2\theta) / 4]}{K(\theta)}$$

4. Находим напряжение вторичной обмотки $U_2=U_o B(\theta)$, пиковое значение тока вентилей (диодов) $I_{\text{вм}}=I_o F(\theta) / m$, действующее значение тока вентилей $I_v=D(\theta) I_o / m$, ток вторичной обмотки $I_2=\sqrt{2} I_v$ и емкость конденсатора фильтра $C=(1 / m+\theta / \pi) / 2 K_{\text{пв}} f R_n$.

Программа 7.59.

```

10 PRINT 'РАСЧЕТ ВЫПРЯМИТЕЛЯ'
20 INPUT 'ВВЕДИТЕ СОПРОТИВЛЕНИЕ НАГРУЗКИ RN=' R1
30 INPUT 'ВВЕДИТЕ ВНУТРЕННЕЕ СОПРОТИВЛЕНИЕ R=' R2
40 INPUT 'ВВЕДИТЕ ЧАСТОТУ F=' F
50 INPUT 'ВВЕДИТЕ ЧИСЛО ФАЗ M=' M
60 INPUT 'ВВЕДИТЕ КОЭФФИЦИЕНТ ПУЛЬСАЦИИ ВЫХОДНОГО НАПРЯЖЕНИЯ K=' K
70 INPUT 'ВВЕДИТЕ НАПРЯЖЕНИЕ U0=' U0
80 INPUT 'ВВЕДИТЕ ТОК I0=' I0
90 INPUT 'ВВЕДИТЕ ТОЧНОСТЬ РЕЗУЛЬТАТА E=' E
100 LET Q=0 : LET H=1
110 IF ABS(H)<E THEN 150
120 LET Q=Q+H : GOSUB 160
130 IF V/H<=0 THEN 120
140 LET H=-H/10 : GOTO 110
150 PRINT 'УГОЛ ОТСЕЧКИ ДИОДОВ Q=' Q : GOTO 180
160 LET Y=TAN(Q)-Q-#PI*R2/M/R1:RETURN
180 LET A=SIN(Q)-Q*COS(Q)
190 LET N=#PI*(1-COS(Q))/A
200 LET B=1/(SQR(2)*COS(Q))
210 LET D=SQR(#PI*(Q*(1+(COS(2*Q))/2)-3*SIN(2*Q)/4))/A
220 LET U=U0*B:LET I1=I0*N/M:LET I2=SQR(2)*I1
230 LET C=((1/M)+(Q/#PI))/(2*K*F*R1)
290 PRINT 'НАПРЯЖЕНИЕ ВТОРИЧНОЙ ОБМОТКИ U2=' U
300 PRINT 'ПИКОВОЕ ЗНАЧЕНИЕ ТОКА ВЕНТИЛЯ I=' I
310 PRINT 'ДЕЙСТВУЮЩЕЕ ЗНАЧЕНИЕ ТОКА ВЕНТИЛЯ I1=' I1
320 PRINT 'ТОК ВТОРИЧНОЙ ОБМОТКИ I2=' I2
330 PRINT 'ЕМКОСТЬ C=' C
340 GOTO 20 : END

```

переменного напряжения питающей сети f , число фаз выпрямителя m , коэффициент пульсаций выходного напряжения $K_{\text{пв}}$ (постоянная составляющая U_o), выходной ток I_o .

2. Из решения трансцендентного уравнения $\text{tg} \theta-\theta=A$, где $A=\pi / m R_n$, находится угол

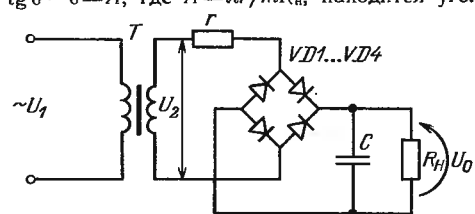


Рис. 7.19. Схема двухфазного (мостового) выпрямителя

Пример. Для $R_n=R_1=1000$ Ом, $r=R_2=100$ Ом, $m=2$, $f=50$ Гц, $K_{\text{пв}}=K=0,1$, $U_o=150$ В, $I_o=0,15$ А и погрешности вычислений $\epsilon=E=1 \cdot 10^{-4}$ получим $\theta=0,7201$ рад, $K(\theta)=K=0,1181$, $B(\theta)=B=0,9406$, $F(\theta)=N=6,602$, $D(\theta)=D=2,294$, $U_2=141,094$ В, $I_{\text{вм}}=0,4952$ А, $I_2=0,2433$ А и $C=7,292 \cdot 10^{-5}$ Ф.

Работу ключевых устройств иногда удается разбить на отдельные стадии и анализировать переходные процессы в них поэтапным методом. Такую возможность рассмотрим на примере инженерного расчета ключа на мощном МДП-транзисторе (см. рис. 7.20, а) (временные диаграммы работы ключа приведены на рис. 7.20, б). Расчет производится по следующему алгоритму.

1. Вводим исходные данные: входную емкость транзистора $C_{\text{вн}}$, проходную емкость

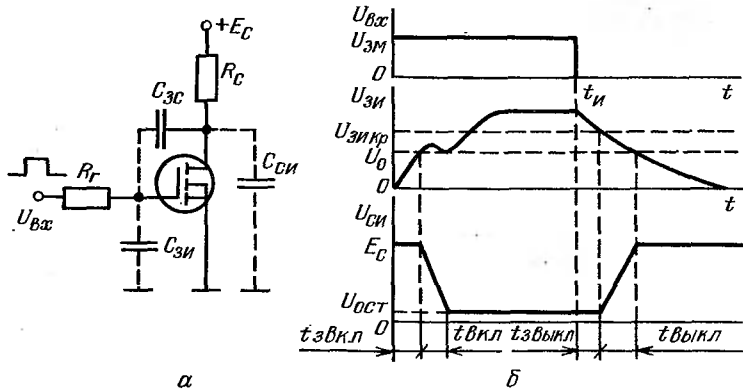


Рис. 7.20. Ключ на мощном U-МДП-транзисторе (а) и временные диаграммы его работы (б)

$C_{ЗС}$, крутизну S_0 , сопротивление сток — исток во включенном состоянии $R_{СИ\text{ вкл}}$, пороговое напряжение отпирания U_0 , напряженне питания E_C , сопротивление нагрузки в цепи стока R_C , сопротивление генератора управляющих импульсов R_G и их амплитуду $U_{ЗМ}$.

2. Находим ток стока во включенном состоянии ключа $I_{CO} = E_C / (R_C + R_{СИ\text{ вкл}})$ и остаточное напряжение на стоке включенного транзистора $U_{ОСТ} = I_{CO} R_{СИ\text{ вкл}}$.

3. Считая, что рабочий участок передаточной характеристики линейный, находим среднее напряжение на затворе в стадии включения $U_{ЗНО} \approx U_0 + \frac{1}{2} \frac{E_C - U_{ОСТ}}{R_C S_0}$ и критическое напряжение на затворе, обеспечивающее включение транзистора $U_{ЗН\text{ кр}} = U_0 + \frac{E_C - U_{ОСТ}}{R_C S_0}$.

4. Находим время задержки включения $T_{ЗВКЛ} = \tau_{ВХЗ} \ln(U_{ЗМ} / (U_{ЗН} - U_0))$, где $\tau_{ВХЗ} = R_G (C_{ЗН} + C_{ЗС})$. Это время определяется зарядом входной емкости $(C_{ЗН} + C_{ЗС})$ закрытого транзистора до напряжения отпирания U_0 .

5. На стадии включения основным фактором, определяющим время включения, оказывается отрицательная обратная связь, осуществляемая через проходную емкость $C_{ЗС}$.

Она ведет к увеличению действующей входной емкости до значения $C_{ВХ\text{ эфф}} = C_{ЗН} + C_{ЗС}(1 + S_0 R_C) \approx C_{ЗС}(1 + S_0 R_C)$, так как $S_0 R_C \gg 1$ и $C_{ЗС}(1 + S_0 R_C) \gg C_{ЗН}$. Возрастание $C_{ВХ}$ ведет к появлению характерного участка медленного изменения напряжения на затворе — оно устанавливается на уровне, близком к $U_{ЗНО}$. Полагая емкость C_{12} линейной, время включения можно найти из выражения $t_{ВКЛ} \approx 0,8 C_{ЗС} \frac{R_G (E_C - U_{ОСТ})}{U_{ЗМ} - U_{ЗНО}}$. Здесь множитель 0,8 учитывает, что в технических расчетах время включения оценивается на уровнях 0,1 и 0,9 от полного перепада напряжения на стоке $(E_C - U_{ОСТ})$.

6. Находим время задержки выключения $t_{ЗВЫКЛ} = \tau_{ВЫКЛ} \ln(U_{ЗМ} / U_{ЗН\text{ кр}})$. Это время определяется экспоненциальным спадом напряжения на затворе от значения $U_{ЗМ}$ до $U_{ЗН\text{ кр}}$.

7. На стадии выключения вновь действует отрицательная обратная связь через проходную емкость $C_{ЗС}$, что ведет к увеличению $C_{ВХ}$ и медленному изменению напряжения на затворе $U_{ЗН}(t) \approx U_{ЗНО}$. Время выключения $t_{ВЫКЛ} \approx 0,8 C_{ЗС} \frac{R_G (E_C - U_{ОСТ})}{U_{ЗНО}}$.

Введение множителя 0,8 обосновано в п. 5 алгоритма.

Программа 7.60.

```

10 PRINT'РАСЧЕТ КЛЮЧА НА МОЩНОМ U-МДП-ТРАНЗИСТОРЕ'
20 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ C11,C12,S0,R ВКЛ,U0 'A,B,S,R,U
30 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ EC,RC,RG 'E,U,Z
40 INPUT'ВВЕДИТЕ АМПЛИТУДУ U ВХМ='U:PRINT'F1.4!''
50 LETI=E/(U+R):LETD=I*R:PRINT'ТОК СТОКА IC ВКЛ='I
60 PRINT'НАПРЯЖЕНИЕ U СИВКЛ='D
70 LETL=(E-D)/U/S:LETF=U+L:LETL=U+L/2
80 LETM=Z*(A+B):LETC=A+B*(1+S*M)
90 PRINT'ЭФФЕКТИВНАЯ ВХОДНАЯ ЕМКОСТЬ C ВХЭФФ='C
100 LETN=C*Z:PRINT'R Г*С ВХЭФФ='N
110 LETP=M*LOG(U/(U-U))
120 PRINT'ВРЕМЯ ЗАДЕРЖКИ ВКЛЮЧЕНИЯ T ЗВКЛ='P
130 LETY=.8*B*Z*(E-D)/(U-L)
140 PRINT'ВРЕМЯ ВКЛЮЧЕНИЯ T ВКЛ='Y:LETX=M*LOG(U/F)
150 PRINT'ВРЕМЯ ЗАДЕРЖКИ ВЫКЛЮЧЕНИЯ T ЗВЫКЛ='X
160 LETT=.8*B*Z*(E-D)/L
170 PRINT'ВРЕМЯ ВЫКЛЮЧЕНИЯ T ВЫКЛ='T:GOTO 40:END

```

Пример. Для $C_{31}=1 \cdot 10^{-9}$ Ф, $C_{32}=30 \cdot 10^{-12}$ Ф, $S_0=1,5$ А/В, $R_{сн\text{ вкл}}=1,5$ Ом, $U_0=6$ В (типовые параметры высоковольтного ключевого U-МДП-транзистора КП701), $E_c=400$ В, $R_c=82$ Ом, $R_r=50$ Ом, $U_{31}=20$ В находим: $I_{c0}=4,79$ А, $U_{сн\text{ вкл}}=7,186$ В, $C_{вх\text{ эф}}=4,72 \cdot 10^{-9}$ Ф, $t_{3\text{ вкл}}=18,37 \cdot 10^{-9}$ с, $t_{вкл}=38 \cdot 10^{-9}$ с, $t_{3\text{ выкл}}=40,03 \cdot 10^{-9}$ с и $t_{выкл}=62,05 \cdot 10^{-9}$ с. Расхождение вычисленных времен задержки и переключения с экспериментальными данными не превышает 15—20 %.

Существенно более высокую точность дает моделирование ключевых устройств методом переменных состояния. Под последними подразумеваются электрические величины, не способные меняться мгновенно (напряжение

$$i_1 = \frac{u_{вх}(t) - u_{сн}}{R_r},$$

$$i_2 = \left(\frac{i_1 - I_c(u_{31}, u_{сн})}{C_{22}} - \frac{i_1}{C_{11}} \right) / \left(\frac{1}{C_{11}} + \frac{1}{C_{22}} + \frac{1}{C_{12}} \right),$$

$$i_3 = i_1 - I_c(u_{31}, u_{сн}) - i_2.$$

Входная емкость C_{11} у мощных U-МДП-транзисторов слабо зависит от напряжения U_{31} , поэтому можно положить $C_{11} = \text{const}$. Однако емкости C_{12} и C_{22} сильно зависят от напряжения $U_{сн}$. В связи с этим данные

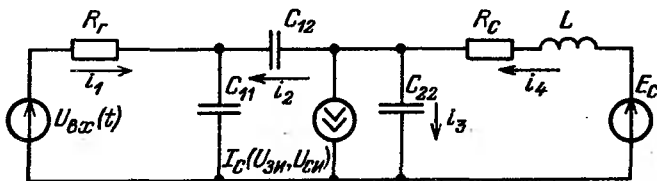


Рис. 7.21. Эквивалентная схема ключа на мощном U-МДП-транзисторе

на емкости, ток в цепи индуктивности и др.). Алгоритм реализации метода переменных состояния заключается в составлении и решении системы нелинейных дифференциальных уравнений состояния, описывающих работу моделируемой схемы.

Например, моделирование ключа (рис. 7.20, а) может выполняться на основе эквивалентной схемы (рис. 7.21). На ней C_{11} — входная, C_{22} — выходная и C_{12} — проходная емкости транзистора, $I_c(U_{31}, U_{сн})$ — источник тока, описываемый выражениями (7.18) и (7.19). Система дифференциальных уравнений для схемы, изображенной на рис. 7.21, имеет вид

$$\frac{du_{31}}{dt} = \frac{i_1 + i_2}{C_{11}},$$

$$\frac{du_{сн}}{dt} = \frac{i_3}{C_{22}},$$

$$\frac{di_4}{dt} = \frac{E_c - i_4 R_c - u_{сн}}{L_c},$$

зависимости аппроксимируются следующими выражениями: $C_{12} = C_{12\text{ мин}} + Ae^{-BU_{сн}}$ и $C_{22} = C_{22\text{ мин}} + Ce^{-DU_{сн}}$, где величины $C_{12\text{ мин}}$, A , B , $C_{22\text{ мин}}$, C и D определяются с помощью экспоненциальной регрессии по известным экспериментальным зависимостям C_{12} и C_{22} от напряжения $U_{сн}$.

Решение данной системы дифференциальных уравнений выполняется методом Рунге — Кутты — Мерсона с автоматическим выбором шага при заданных начальных условиях $U_{31}(t_0)$, $U_{сн}(t_0)$, $i_4(t_0)$ и t_0 , а также заданной временной зависимости $U_{вх}(t)$. В приведенной ниже программе $U_{вх}(t)$ задается в виде импульса с экспоненциальным нарастанием и спадом

$$u_{вх}(t) = U_{м\text{ вх}}(1 - e^{-t/\tau_{вх}}) \quad \text{при } 0 \leq t \leq t_n,$$

$$u_{вх}(t) = U_{м\text{ вх}}(1 - e^{-t_n/\tau_{вх}}) e^{-(t-t_n)/\tau_{вх}} \quad \text{при } t > t_n,$$

где $U_{м\text{ вх}}$ — амплитуда, t_n — длительность и $\tau_{вх}$ — постоянная времени нарастания и спада входного импульса.

Программа 7.61.

```

10 PRINT'МОДЕЛИРОВАНИЕ КЛЮЧА НА МОЩНОМ U-МДП-ТРАНЗИСТОРЕ'
15 DIM Y(3),W(3),A(3),C(3),D(3),E(3),F(3)
20 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ IS,US,S,K 'I,U,S,P
30 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ C11,C12 МИН,А,В 'C,A9,B9,C9
40 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ C22 МИН,С,Д 'D9,E9,F9
50 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ EC,RC,RG,L 'E,Z,G,A
60 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ U MEX,T BX,T И 'U,T,F
70 INPUT'ВВЕДИТЕ ПАРАМЕТРЫ H0,EPS,T0 'H,E1,X,LETN=3
90 LETY(1)=0:LETW(1)=0:LETY(2)=E:LETW(2)=E:LETY(3)=0:LETW(3)=0
110 LETE3=0:GOSUB 400:LETD=0:FOR J=1 TO N
120 LETA(J)=F(J)*H:LETY(J)=W(J)+A(J)/3:NEXT J
130 LETX=X+H/3:GOSUB 400:FOR J=1 TO N

```



```

140 LETY(J)=W(J)+A(J)+F(J)*H)/6:NEXT J
150 GOSUB 400:FOR J=1 TO N:LETC(J)=F(J)*H
160 LETY(J)=W(J)+A(J)/8+.375*C(J):NEXT J
170 LETX=X+H/6:GOSUB 400:FOR J=1 TO N
180 LETD(J)=F(J)*H:LETY(J)=W(J)+A(J)/2-1.5*C(J)+2*D(J)
190 NEXT J:LETX=X+H/2:GOSUB 400
200 FOR J=1 TO N:LETE(J)=F(J)*H
210 LETY(J)=W(J)+(A(J)+4*D(J)+E(J))/6
220 LETE2=ABS(-2*A(J)+9*C(J)-8*D(J)+E(J))/30
230 IF E2<=E1 THEN 250
240 LETE3=1:GOTO 260
250 IF E2<E1/20 THEN LETD=D+1
260 NEXT J:IF E3=0 THEN 290
270 LETX=X-H:FOR J=1 TO N:LETY(J)=W(J):NEXT J
280 LETH=H/2:GOTO 110
290 IF D=N THEN LETH=H+H
300 PRINT:IF 1.21'T='X,'U BX='A7,'U 3='Y(1),'U C='Y(2)
310 FOR J=1 TO N:LETW(J)=Y(J):NEXT J:GOTO 110
400 IF X<F THEN LETB7=U*(1-EXP(-X/T))
410 LETA7=B7:IF X>F THEN LETA7=B7*EXP(-(X-F)/T)
420 LETD8=A7+B7*EXP(-C9*Y(2)):LETE8=D9+E9*EXP(-F9*Y(2))
440 LETF8=I*(1+HTN(S*(Y(1)-U)/I))
450 LETY=F8*(1-EXP(-P*S*Y(2)/F8)):LETA8=(A7-Y(1))/6
470 LETB8=((Y(3)-Y)/E8-A8/C)/(1/C+1/D8+1/E8)
480 LETC8=Y(3)-Y-B8:LETF(1)=(A8+B8)/C:LETF(2)=C8/E8
490 LETF(3)=(E-Y(3))*Z-Y(2)/A:RETURN:END

```

Пример. Для $I_s=11$ А, $U_s=12,6$ В, $S=2,72$ А/В, $K=0,7$, $C_{11}=3,5 \cdot 10^{-10}$ Ф, $C_{12 \text{ мин}}=8 \cdot 10^{-12}$ Ф, $A=3,5 \cdot 10^{-11}$ Ф, $B=0,088$ 1/В, $C_{22 \text{ мин}}=1,3 \cdot 10^{-10}$ Ф, $C=6,59 \cdot 10^{-10}$ Ф, $D=0,33$ 1/В (параметры транзистора КП913А), $E_c=45$ В, $R_c=5$ Ом, $L=1 \cdot 10^{-8}$ Ф, $U_{\text{нвб}}=20$ В, $\tau_{\text{вх}}=3 \cdot 10^{-9}$ с, $t_{\text{н}}=14 \cdot 10^{-9}$ с, $h_{\alpha}=1 \cdot 10^{-9}$ с (начальный шаг), $Q=1 \cdot 10^{-3}$ (погрешность вычислений) и $t_0=0$ получаем следующие данные:

двух дифференциальных уравнений

$$\frac{di}{dt} = \frac{E - iR - u}{L}, \quad \frac{du}{dt} = \frac{i - I(u)}{C},$$

где $I(U)$ — зависимость тока нелинейного двухполюсника от приложенного напряжения U . Отметим, что часть общей емкости C отражает инерционность двухполюсника.

t , нс	0,5	1,5	2,5	3	3,5	4	4,5	5	5,5	6	...
$u_{\text{вх}}(t)$, В	3,07	7,87	11,3	12,6	13,8	14,7	15,5	16,2	16,8	17,3	...
$u_{\text{ан}}(t)$, В	0,139	1,07	2,53	3,37	4,23	5,1	5,96	6,79	7,57	8,31	...
$u_{\text{сн}}(t)$, В	44,9	44,6	44,2	43,9	43,3	42,4	41	38,9	35,9	31,8	...

Зависимости $u_{\text{вх}}(t)$, $u_{\text{ан}}(t)$ и $u_{\text{сн}}(t)$, рассчитанные по программе 7.61, показаны на рис. 7.22, причем на графике зависимости $U_{\text{сн}}(t)$ отмечены точки — результаты вычислений. Из расположения этих точек отчетливо виден характер автоматического изменения шага интегрирования дифференциальных уравнений — на крутых участках $U_{\text{сн}}(t)$ шаг автоматически уменьшается, а на пологих — возрастает. В общей сложности для построения этих кривых понадобилось 72 цикла вычислений. Если бы использовался обычный метод Рунге — Кутты с шагом $h=0,25 \cdot 10^{-9}$ с (такой шаг нужен для получения погрешности $Q \leq 1 \cdot 10^{-3}$), то при общем временном интервале $30 \cdot 10^{-9}$ с потребовалось бы 120 циклов вычислений. При увеличении протяженности пологих участков зависимости $U_{\text{сн}}(t)$ выигрыш от применения метода Рунге — Кутты — Мерсона становится еще заметнее.

Моделирование нелинейной автоколебательной цепи (рис. 7.23) с двухполюсником, имеющим N -образную вольт-амперную характеристику, сводится к решению системы из

Допустим, что N -образная вольт-амперная характеристика двухполюсника (в нашем случае туннельного диода 3И202К) задана графически (рис. 7.24). Аналитически ее можно представить полиномом 5-й степени, записанным по схеме Горнера

$$I(u) = (((((a_5 u + a_4) u + a_3) u + a_2) u + a_1) u + a_0. \quad (7.21)$$

Для нахождения коэффициентов a_0 — a_5 полинома составим таблицу значений u и I :

u , В	0	0,2	0,4	0,6	0,8	1,0
I , мА	0	52	23	2	4	13

Используя любую из описанных ранее программ для полиномиальной интерполяции и аппроксимации при $n=5$ и $h=\Delta U=0,2$ В, находим: $a_0=0$, $a_1=746,333$, $a_2=-3420,833$, $a_3=5734,375$, $a_4=-4166,667$ и $a_5=1119,792$. Кривая, построенная по формуле (7.21), практически точно совпадает с исходным

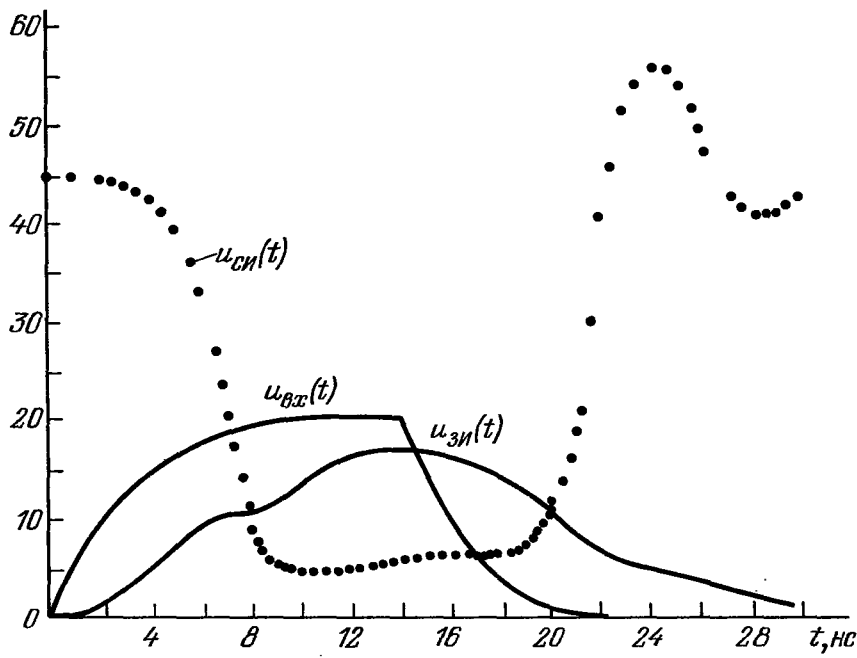
$u_{сн}, u_{зн}, u_{вх}$ 

Рис. 7.22. Расчетные зависимости $u_{вх}(t)$, $u_{зн}(t)$ и $u_{сн}(t)$ для ключа на мощном U -МДП-транзисторе, полученные при его моделировании

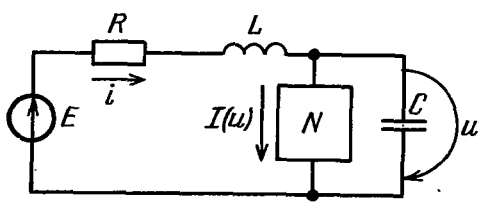


Рис. 7.23. Нелинейная автоколебательная схема с двухполюсником, имеющим N -образную вычислительную характеристику

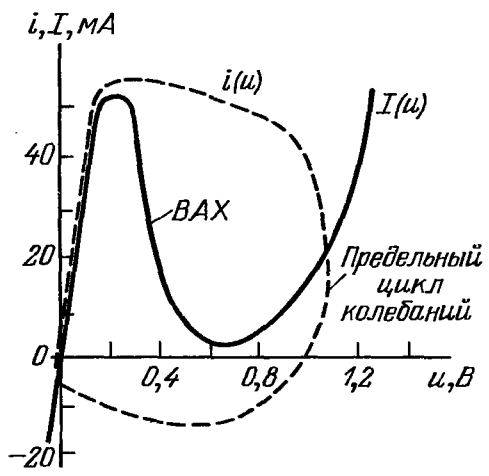


Рис. 7.24. Фазовый портрет автоколебаний, построенный в плоскости вольт-амперной характеристики

графиком зависимости $I(u)$, что свидетельствует о высокой точности аппроксимации.

Для решения приведенной выше системы воспользуемся методом Рунге — Кутты —

Мерсона, дополнительно организовав ввод исходных данных и подпрограмму для вычисления $I(u)$, $di/dt = F0$ и $du/dt = F1$ при $i = A0$ и $u = A1$.

```

10 PRINT 'МОДЕЛИРОВАНИЕ РЕЛАКСАЦИОННОГО ГЕНЕРАТОРА '
20 PRINT 'НА ПРИБОРЕ С N-ОБРАЗНОЙ В.А.Х.'
25 LETN=2: DIM Y(N), W(N), A(N), C(N), D(N), E(N), F(N)
30 PRINT 'ВВЕДИТЕ КОЭФФИЦИЕНТЫ АППРОКСИМАЦИИ В.А.Х.'
40 INPUT 'A0, A1, A2, A3, A4, A5', Q0, Q1, Q2, Q3, Q4, Q5
50 INPUT 'ВВЕДИТЕ ПАРАМЕТРЫ СХЕМЫ E, R, C, L', E, Z, C, L
60 INPUT 'ЗАДАЙТЕ НАЧАЛЬНЫЙ ШАГ H0=' H
70 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ EPS=' E1
90 LETY(1)=0: LETW(1)=0: LETY(2)=0: LETW(2)=0: LETX=0
110 LETE3=0: GOSUB 400: LETD=0: FOR J=1 TO N
120 LETA(J)=F(J)*H: LETY(J)=W(J)+A(J)/3: NEXT J
130 LETX=X+H/3: GOSUB 400: FOR J=1 TO N
140 LETY(J)=W(J)+(A(J)+F(J)*H)/6: NEXT J
150 GOSUB 400: FOR J=1 TO N: LETC(J)=F(J)*H
160 LETY(J)=W(J)+A(J)/8+.375*C(J): NEXT J
170 LETX=X+H/6: GOSUB 400: FOR J=1 TO N
180 LETD(J)=F(J)*H: LETY(J)=W(J)+A(J)/2-1.5*C(J)+2*D(J)
190 NEXT J: LETX=X+H/2: GOSUB 400
200 FOR J=1 TO N: LETE(J)=F(J)*H
210 LETY(J)=W(J)+(A(J)+4*D(J)+E(J))/6
220 LETE2=ABS(-2*A(J)+9*C(J)-8*D(J)+E(J))/30
230 IF E2<=E1 THEN 250
240 LETE3=1: GOTO 260
250 IF E2<E1/20 THEN LETD=D+1
260 NEXT J: IF E3=0 THEN 290
270 LETX=X-H: FOR J=1 TO N: LETY(J)=W(J): NEXT J
280 LETH=H/2: GOTO 110
290 IF D=N THEN LETH=H+H
300 PRINT '1.2! T=' X, 'U=' Y(2), 'IR=' Y(1)*I E3
310 LETW(1)=Y(1): LETW(2)=Y(2): GOTO 110
400 LETY(2): LETI=(((Q5*Y+Q4)*Y+Q3)*Y+Q2)*Y+Q1)*Y+Q0
410 LETF(1)=(E-Y(1)*Z-Y(2))/L
420 LETF(2)=(Y(1)-I/IE3)/C: RETURN: END

```

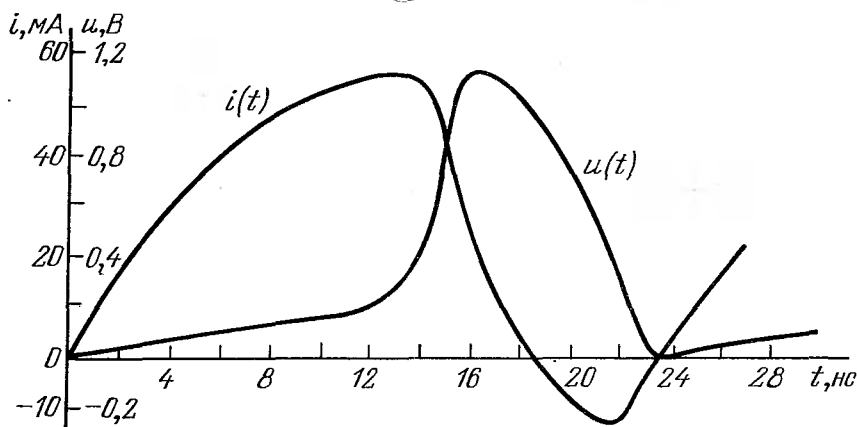


Рис. 7.25. Расчетные временные зависимости $u(t)$ и $i(t)$ для автоколебательной схемы на туннельном диоде

Пример. Для указанных значений $a_0 - a_1$, $E = 0,5$ В, $R = 5$ Ом, $C = 100 \cdot 10^{-12}$ Ф, $L = 50 \cdot 10^{-9}$ Гн, $h_0 = 1 \cdot 10^{-9}$ с (начальный шаг), $Q = 1 \cdot 10^{-3}$ (погрешность вычислений) и $t_0 = 0$ будем получать следующие результаты:

На рис. 7.24 показана фазовая траектория автоколебаний при нулевых начальных условиях, а на рис. 7.25 — форма импульсов напряжения $u(t)$ и тока $i(t)$. Аналогичные расчетным данные получены при экспериментальном исследовании данной схемы.

$t \times 10^{-9}$, с	0,25	0,5	1	1,5	2	3	3,5	4	5	...
u , В	0,00179	0,00488	0,0113	0,018	0,0246	0,0365	0,0444	0,051	0,0638	...
i , мА	1,53	3,59	8,3	12,4	16,5	24,7	26,5	29,9	35,9	...
i , мА	2,47	4,86	9,42	13,7	17,7	24,9	28,2	31,2	36,7	...

§ 7.8. Расчёты в механике и термодинамике

Рассмотрим некоторые примеры расчета параметров движущихся тел [21]. При частично упругом соударении (центрально-лобовом) двух тел с массами m_1 и m_2 и скоростями v_1 и v_2 скорости после соударения

$$v_1' = \frac{m_1 v_1 + m_2 v_2 - (v_1 - v_2) m_2 k}{m_1 + m_2},$$

$$v_2' = \frac{m_1 v_1 + m_2 v_2 + (v_1 - v_2) m_1 k}{m_1 + m_2}$$

и энергия, теряемая при соударении,

$$W = \frac{m_1 m_2}{2(m_1 + m_2)} (v_1 - v_2)^2 (1 - k^2),$$

где k — коэффициент восстановления.

Программа 7.63.

```
10 PRINT'РАСЧЕТ СКОРОСТЕЙ И ПОТЕРИ ЭНЕРГИИ ПРИ ЛОБОВОМ'
20 PRINT' СТОЛКНОВЕНИИ ДВУХ ТЕЛ'
30 INPUT'ВВЕДИТЕ МАССЫ ТЕЛ M1,M2 'A,B
40 INPUT'ВВЕДИТЕ СКОРОСТИ ТЕЛ ДО СОУДАРЕНИЯ U1,U2 'C,D
50 INPUT'ВВЕДИТЕ КОЭФФИЦИЕНТ ВОССТАНОВЛЕНИЯ K='K
60 LETE=A*C+V*B:D:LETM=A+V:LETU=(C-D)*K
70 LETU1=(E-U*B)/M:LETU2=(E+U*B)/M
75 LETW=A*B*(C-D)^2*(1-K*K)/2/M
80 PRINT'СКОРОСТЬ U1 ПОСЛЕ СОУДАРЕНИЯ ='U1
90 PRINT'СКОРОСТЬ U2 ПОСЛЕ СОУДАРЕНИЯ ='U2
100 PRINT'ПОТЕРЯ ЭНЕРГИИ ПОСЛЕ УДАРА W='W:END
```

Пример. Для $m_1=1$ кг, $m_2=1$ кг, $v_1=2$ м/с, $v_2=1$ м/с, $k=5/9$ получим $v_1'=1,222$ м/с, $v_2'=1,778$ м/с и $W=0,1728$ Дж.

Тело массой m , брошенное вверх под углом к горизонту α с начальной скоростью v_0 , поднимается на максимальную высоту h_m за время $t_{h_m}=(v_0 \sin \alpha)/g$ при общем времени движения (до падения) $t_{sm}=2t_{h_m}$. При этом максимальная высота подъема $h_m=(v_0^2 \sin^2 \alpha)/2g$, а дальность полета $s_m=(v_0^2 \sin 2\alpha)/g$. В заданный момент времени t расстояние s и высота h определяются параметрически заданными уравнениями $s=v_0 t \cos \alpha$, $h=v_0 t \sin \alpha - gt^2/2$, где g — ускорение свободного падения. Скорость тела при этом $v=\sqrt{v_0^2 - 2gh}$.

Программа 7.64.

```
05 PRINT'РАСЧЕТ ПАРАМЕТРОВ ДВИЖЕНИЯ ТЕЛА, БРОШЕННОГО ПОД'
10 PRINT'УГЛОМ Q К ГОРИЗОНТУ С НАЧАЛЬНОЙ СКОРОСТЬЮ U0,'
15 PRINT' БЕЗ УЧЕТА СОПРОТИВЛЕНИЯ ВОЗДУХА'
20 INPUT'ВВЕДИТЕ НАЧАЛЬНУЮ СКОРОСТЬ U0='U
30 INPUT'ВВЕДИТЕ УГОЛ Q В ГРАДУСАХ Q='Q
40 INPUT'ВВЕДИТЕ УСКОРЕНИЕ СВОБОДНОГО ПАДЕНИЯ G='G
50 LETQ=RAD(Q):LETT=U0*SIN(Q)/G
60 PRINT'ВРЕМЯ ДВИЖЕНИЯ ДО МАКСИМАЛЬНОЙ ВЫСОТЫ T МАКС='T
70 PRINT'ОБЩЕЕ ВРЕМЯ ДВИЖЕНИЯ T SM='2*T
80 PRINT'МАКСИМАЛЬНАЯ ВЫСОТА H='U0*U0*SIN(Q)^2/G
90 PRINT'ДАЛЬНОСТЬ ПОЛЕТА S M='U0*U0*SIN(2*Q)/G
100 INPUT'ВВЕДИТЕ ВРЕМЯ T='T
110 PRINT'РАССТОЯНИЕ S(T)='U0*T*COS(Q)
120 LETH=U0*T*SIN(Q)-G*T^2/2:PRINT'ВЫСОТА H(T)='H
130 PRINT'СКОРОСТЬ U(T)='SQR(U0*U0-2*G*T*H)
140 GOTO 100:END
```

Пример. Для $v_0=130$ м/с, $\alpha=Q=25^\circ$, $G=g=9,8$ м/с получаем $t_{h_m}=3,08, t_{sm}=6,16$ с, $h_m=H(M)=308,005$ м, $s_m=S(M)=1321,036$ м. Далее, задав $t=T=0,5$ с, получим $S(T)=58,91$ м, $H(T)=28,245$ м и $U(T)=28,006$ м/с.

В приведенном выше примере не учитывалось сопротивление воздуха, существенное при больших скоростях. При учете сопротивления воздуха полет тела описывается системой из двух нелинейных дифференциальных уравнений, решаемых методом Рунге — Кутты или другим методом (см. § 4.10).

Период колебаний реального тела (физического маятника) с распределенной массой m и моментом инерции I_A , подвешенного или подвижно закрепленного на точке A , не совпадающей с положением центра масс (точка O), определяется выражением $T=2\pi\sqrt{I_A/mgs}$, где s — расстояние от точки O до точки A . Соответственно, зная период

колебаний физического маятника, можно найти его момент инерции относительно точки $I_0=ms\left(\frac{gT^2}{4\pi^2}-s\right)$. Программа 7.65

обеспечивает расчет T или I_0 (искомой величины при вводе задается значение 0).

Программа 7.65.

Пример. Для $m=0,5$ кг, $s=0,5$ м, $g=9,8$ м/с и $I_A=2$ кг/м, задав $T=0$, получим $T=5,67690758$ с. Задав $T=5$ с и $I=0$, получим $I_0=1,426480625$ кг/м².

Атмосферное давление P (в килопаскалях) и плотность воздуха R (кг/м³) на высоте H (км) определяется формулами

$$P=101,3(1-6,5H/288)^{5,255},$$

$$R=1,2255(1-6,5H/288)^{4,255}.$$

Программа 7.65.

```

10 PRINT 'РАСЧЕТ ПАРАМЕТРОВ ФИЗИЧЕСКОГО МАЯТНИКА'
20 PRINT 'НЕИЗВЕСТНОМУ ЗНАЧЕНИЮ I ИЛИ T ЗАДАЙТЕ ЗНАЧЕНИЕ Ø'
30 INPUT 'ВВЕДИТЕ МАССУ M='M
35 INPUT 'ВВЕДИТЕ УСКОРЕНИЕ СВОБОДНОГО ПАДЕНИЯ G='G
40 INPUT 'РАССТОЯНИЕ ОТ ЦЕНТРА ТЯЖЕСТИ ДО ТОЧКИ ПОДВЕСА S='S
50 INPUT 'ЗАДАЙТЕ МОМЕНТ ИНЕРЦИИ (ИЛИ Ø) I='I
60 INPUT 'ЗАДАЙТЕ ПЕРИОД КОЛЕБАНИЙ (ИЛИ Ø) T='T
70 IF I=Ø THEN GØ
80 PRINT 'ПЕРИОД КОЛЕБАНИЙ МАЯТНИКА T='2*#PI*S*SQR(1/M/S/G):GOTO 50
90 PRINT 'МОМЕНТ ИНЕРЦИИ IØ='M*S*(G*T*T/4/#PI^2-S):GOTO 50:END
    
```

Давление P , выраженное в миллиметрах ртутного столба, получаем умножением P на 7,5.

Программа 7.66.

```

10 PRINT 'АТМОСФЕРНОЕ ДАВЛЕНИЕ И ПЛОТНОСТЬ ВОЗДУХА'
20 INPUT 'ВВЕДИТЕ ВЫСОТУ H(КМ)='H
30 LETP=101.3*(1-6.5*H/288)^5.255
40 PRINT!F3.4!'ДАВЛЕНИЕ P='P'КИЛОПАСКАЛЬ'
50 PRINT!F3.4!'ДАВЛЕНИЕ P='P*7.5'ММ РТУТНОГО СТОЛБА'
60 LETR=1.225*(1-6.5*H/288)^4.255
70 PRINT!F1.5!'ПЛОТНОСТЬ ВОЗДУХА R='R' КГ/М^3'
80 GOTO 20:END
90 PRINT 'СКОРОСТЬ U2 ПОСЛЕ СОУДАРЕНИЯ ='U2
100 PRINT 'ПОТЕРЯ ЭНЕРГИИ ПОСЛЕ УДАРА W='W:END
    
```

Максимальное число слоев при реализации этого алгоритма определяется максимальным номером индекса i для одномерного массива, отведенного под L_i/K_i .

Пример. Для $H=1$ км получаем $P=89,84852$ кПа, $P=673,86387$ мм рт. столба и $R=1,11206$ кг/м³.

Распределение температуры в стационарных условиях в многослойном стержне (рис. 7.26) может вычисляться по следующему алгоритму.

1. Вводим исходные данные (число слоев N , температуру свободной поверхности первого слоя T_0 и последнего T_n , площадь поперечного сечения стержня s , толщину слоев L_1-L_n и коэффициенты термической проводимости K_1-K_n). По мере ввода L_i, K_i вычисляем и запоминаем значения L_i/K_i .

2. Вычисляем величину $Q = (T_0 - T_n) / \sum_{i=1}^N (L_i/K_i) / s$, организовав с помощью

цикла вызов значений L_i/K_i при $i=1, 2, \dots, n$.

3. С помощью рекуррентного соотношения $T_i = T_{i-1} - Q (L_i/K_i) / s$ находим температуру каждого слоя.

Программа 7.67.

```

10 PRINT 'РАСЧЕТ СТАЦИОНАРНОГО РАСПРЕДЕЛЕНИЯ ТЕМПЕРАТУРЫ'
20 PRINT ' ПО ДЛИНЕ МНОГОСЛОЙНОГО СТЕРЖНЯ'
30 INPUT 'ВВЕДИТЕ ЧИСЛО СЛОЕВ N='N:DIM P(N)
40 INPUT 'ВВЕДИТЕ ТЕМПЕРАТУРУ В НАЧАЛЕ СТЕРЖНЯ T0='T
50 INPUT 'ВВЕДИТЕ ТЕМПЕРАТУРУ В КОНЦЕ СТЕРЖНЯ TN='W
60 INPUT 'ВВЕДИТЕ ПЛОЩАДЬ ПОПЕРЕЧНОГО СЕЧЕНИЯ СТЕРЖНЯ S='S
70 PRINT 'ВВЕДИТЕ ДЛИНУ СЛОЕВ И ИХ ТЕРМИЧЕСКУЮ ПРОВОДИМОСТЬ'
80 LETR=Ø:FOR I=1 TO N
90 PRINT!3.Ø!'L('I'),K('I')':INPUT 'L,K
100 LETP(I)=L/K:LETR=R+P(I):NEXT I
110 LETQ=(T-W)*S/R:PRINT!F1.9!'Q='Q
120 LETH=T:FOR I=1 TO N:LETH=H-Q*P(I)/S
130 PRINT!3.Ø!'ТЕМПЕРАТУРА T'I'='!4.6!H:NEXT I:END
    
```

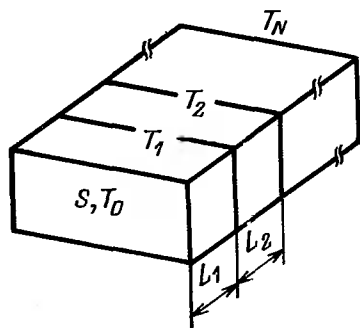


Рис. 7.26. К расчету стационарного распределения температуры по длине многослойного стержня

Пример. Для трехслойного стержня, имеющего $n=3$, $s=1$ м², $L_1=0,25$ м, $K_1=1,2$ (K_i измеряется в ккал/м·час·К°), $L_2=0,12$ м, $K_2=0,25$, $L_3=0,15$, $K_3=0,75$, $T_0=900$ °С и $T_3=100$ °С получим $T_1=712,383$ °С и $T_2=280,113$ °С.

§ 7.9. Финансово-экономические расчеты

Сумма s , получаемая при $P\%$ годовых начислений и первоначальном единовременном вкладе s_0 за период N , определяется по формуле сложных процентов

$$s = s_0 (1 + P/100)^N.$$

Из этой формулы получаем следующие выражения:

$$s_0 = s / (1 + P/100)^N, \quad N = \frac{\lg s - \lg s_0}{\lg (1 + P/100)},$$

$$P = 100 [(s/s_0)^{1/N} - 1].$$

Зная три заданных параметра, по этим формулам находится четвертый — неизвестный параметр. В программе 7.75 для нахождения искомого параметра при его вводе задаем ему нулевое значение.

Программа 7.68.

```

10 PRINT 'ВЫЧИСЛЕНИЯ СО СЛОЖНЫМИ ПРОЦЕНТАМИ'
20 PRINT 'ИСКОМОЙ ВЕЛИЧИНЕ ЗАДАЙТЕ НУЛЕВОЕ ЗНАЧЕНИЕ'
30 INPUT 'ВВЕДИТЕ КОНЕЧНУЮ ВЕЛИЧИНУ S=' S
40 INPUT 'ВВЕДИТЕ НАЧАЛЬНУЮ ВЕЛИЧИНУ S0=' Z
50 INPUT 'ВВЕДИТЕ ПРОЦЕНТ P=' P
60 INPUT 'ВВЕДИТЕ ЧИСЛО ПЕРИОДОВ N=' N
70 PRINT!6.2!' : LET A=1+P/100
80 IF S=0 THEN PRINT'S='Z*A^N
90 IF Z=0 THEN PRINT'S0='S/A^N
100 IF N=0 THEN PRINT'N='LOG(S/Z)/LOG(A)
110 IF P=0 THEN PRINT'P='((S/Z)^(1/N)-1)*100
120 END
    
```

Пример А. Пользователь сберегательной кассой внес в нее вклад $s_0=5000$ руб. До какой суммы s он возрастет через $N=5$ лет, если процент годовых начислений $P=3\%$. Вводим $s=0$, $s_0=5000$, $N=5$, $P=3$, получаем $s=5796,37$ руб.

Пример В. Вскоре после рождения сына родители решили внести в сберкассу вклад s_0 с тем, чтобы через 22 года (предполагаемое время его женитьбы) вклад вырос до суммы $s=30000$ руб., достаточной для приобретения дома. Каков должен быть вклад s_0 , если процент годовых начислений $P=3\%$? Вводим $s=30000$, $s_0=0$, $N=22$, $P=3$, получаем $s_0=15656$ руб. (т. е. почти вдвое меньше s).

Пример С. Через сколько лет начальный вклад $s_0=20000$ руб. увеличится до суммы $s=30000$ руб., если процент годовых начислений $P=3\%$? Вводим $s=30000$, $s_0=20000$, $N=0$, $P=3$, получаем $N=13,72$ года (примерно 14 лет).

Пример D. Какой процент годовых начислений P должен обеспечить банк, чтобы первоначальный вклад $s_0=15000$ руб. увеличился вдвое за время $N=18$ лет? Вводим $s=2 \cdot 15000=30000$, $s_0=15000$, $N=18$, $P=0$, получаем $P=3,926\%$.

Если в каждый из N периодов (лет или месяцев) вкладывается (или погашается) сумма s_1 при $P\%$ годовых начислений, то общая сумма вклада (погашения) составит величину

$$s = s_1 \frac{(1+I)^N - 1}{I},$$

где

$$I = P/100.$$

Из этой формулы получим

$$N = \frac{\ln(sI/s_1 + 1)}{\ln(1+I)} \quad \text{и} \quad s_1 = \frac{sI}{(1+I)^N - 1}.$$

Однако относительно I или P явного выражения получить не удастся. Поэтому нахождение $P=I \cdot 100$ осуществляется численным методом поразрядного приближения из решения нелинейного уравнения

$$F(I) = sI / [(1+I)^N - 1] - s_1 = 0.$$

Выбор неизвестного параметра (s , N , s_1 или P) обеспечивается заданием ему нулевого значения при вводе.

Программа 7.69.

```

10 PRINT 'РАСЧЕТЫ ВКЛАДА (ПОГАШЕНИЯ) ПРИ ПЕРИОДИЧЕСКИХ'
20 PRINT ' РАВНЫМ ВЗНОСАМ'
30 PRINT 'ИСКОМОЙ ВЕЛИЧИНЕ ЗАДАЙТЕ НУЛЕВОЕ ЗНАЧЕНИЕ'
40 INPUT 'ВВЕДИТЕ КОНЕЧНУЮ ВЕЛИЧИНУ S=' S
50 INPUT 'ВВЕДИТЕ СУММУ КАЖДОГО ВКЛАДА S1=' Z
60 INPUT 'ВВЕДИТЕ ПРОЦЕНТ P=' P
70 INPUT 'ВВЕДИТЕ ЧИСЛО ВКЛАДОВ N=' N
80 PRINT!8.2!' : LET I=P/100
90 IF S=0 THEN PRINT'S='Z*((1+I)^N-1)/I
100 IF N=0 THEN PRINT'N='LOG(S*I/Z+1)/LOG(1+I)
110 IF Z=0 THEN PRINT'S1='S*I/((1+I)^N-1)
115 IF P<>0 THEN STOP
120 LET I=0:LETH=.01
130 LET I=I+H:GOSUB 170
140 IF F/H>=0 THEN 130
150 LET H=-H/4:IF ABS(H)>2.5E-4 THEN 130
160 PRINT'P='100*I ' %':STOP
170 LET F=S*I/((1+I)^N-1)-Z:RETURN:END
    
```

Пример А. Вкладчик сберегательной кассы решил ежегодно вносить вклад $s_1 = 1000$ руб. в течение $N=10$ лет. Сколько денег он сможет получить, если процент годовых начислений $P=3\%$? Вводим $s=0$, $s_1=1000$, $P=3$, $N=10$, получаем $s=11463,88$ руб.

Пример В. Семья решила накопить для покупки дома сумму $s=30000$ руб.,

Сортировка заключается в вычислении общей суммы данных, сумм по категориям и процентов (от общей суммы) по категориям. Алгоритм сортировки сводится к указанию номера элемента одномерного массива N и выполнению на каждом цикле ввода операций $s \leftarrow s + X$ (вычисление общей суммы) и $A(N) \leftarrow A(N) + X$ (вычисление суммы по N -й категории).

Программа 7.70.

```

10 PRINT 'СОРТИРОВКА ДАННЫХ ПО КАТЕГОРИЯМ'
20 PRINT 'ДЛЯ ВЫДАЧИ РЕЗУЛЬТАТОВ ЗАДАЙТЕ N=X=0'
30 INPUT 'ВВЕДИТЕ N МАКС='M':DIM A(M):LETS=0
40 FOR I=0 TO M:LETA(I)=0:NEXT I
50 INPUT 'ВВЕДИТЕ N, X 'N,X
60 IF N=0 THEN 80
70 LETA(N)=A(N)+X:LETS=S+X:GOTO 50
80 FOR I=1 TO M
90 PRINT!3.0!'S('I')='!8.2!A(I)
100 PRINT!2.2!'P(%)='A(I)*100/S:NEXT I
110 PRINT!8.2!'ОБЩАЯ СУММА S0='S:END

```

вкладывая в сберкассу ежегодно сумму $s_1 = 10000$ руб. Сколько лет N на это потребуются, если процент годовых начислений $P=3\%$? Вводим $s=30000$, $s_1=10000$, $P=3$, $N=0$, получаем $N=21,71$.

Пример С. Какую сумму s_1 должен ежегодно вносить вкладчик, чтобы через $N=20$ лет накопить сумму $s=30000$ руб. при проценте годовых $P=3\%$? Вводим $s=30000$, $s_1=0$, $P=3$ и $N=20$, получаем $s_1=1116,47$ руб.

Пример D. Какой должен быть процент годовых начислений P , чтобы при ежегодном вкладе $s_1=1000$ руб. накопить сумму $s=30000$ руб. за $N=20$ лет? Вводим $s=30000$, $s_1=1000$, $P=0$, $N=20$, получаем $P=4,13\%$.

К распространенным финансово-экономическим расчетам относится сортировка данных по категориям.

Например, данные могут быть представлены в виде месячной заработной платы рабочих со специальностями различных разрядов (категорий) в виде следующей таблицы:

Пример. Введя $N_{\text{макс}}=6$ и данные приведенной выше таблицы (значения N и X), по окончании ввода указываем $N=0$ и $X=0$. Получаем результат сортировки:

Категория N	1	2	3
$S(N)$	153	175	397
$P(\%)$	6,75	7,72	17,51

4	5	6	Итого
457	700	385	2267
20,16	30,88	19,68	100

Максимальное число категорий $N_{\text{макс}}$ определяется максимальным номером индекса у одномерных массивов. Ограничений на $M_{\text{макс}}$ нет.

Часто возникает необходимость в суммировании табличных данных по горизонтальным или вертикальным таблицам. Если данные вводятся построчно, то при M столбцах и N строках для такой обработки необходим одномерный массив из M суммирующих регистров. С его помощью производится суммирование по столбцам.

Рабочий (фамилия)	Иванов	Петров	Сидоров	Козлов	Волков	Петров	Серов	Васяленко
Категория (разряд)	2	5	6	1	3	3	5	1
Месячный заработок (руб.)	85	175	190	75	135	130	170	78

Рабочий (фамилия)	Карпов	Хохлов	Иваненко	Голубев	Птицын	Зверев	Простак-ков	Левашов
Категория (разряд)	4	6	5	4	3	4	2	5
Месячный заработок (руб.)	150	195	175	155	132	152	90	180

Программа 7.71.

```

10 PRINT 'СУММИРОВАНИЕ ДАННЫХ ТАБЛИЦЫ ПО ГОРИЗОНТАЛЯМ И ВЕРТИКАЛЯМ'
20 INPUT 'ВВЕДИТЕ ЧИСЛО СТРОК И СТОЛБЦОВ M,N,M,N:DIM A(N)'
30 FOR X=1 TO N:LET A(X)=0:NEXT X:LETT=0
40 FOR Y=1 TO M:LETS=0:FOR X=1 TO N
50 PRINT!3.0!'ВВЕДИТЕ Y'Y',X'X:INPUT''D
60 LETS=S+D:LETT=T+D:LET A(X)=A(X)+D:NEXT X
70 PRINT!3.0!'СУММА Y'Y!8.2!'='S:NEXT Y
80 FOR X=1 TO M:PRINT!3.0!'СУММА X'X!8.2!'='A(X):NEXT X
90 PRINT 'ОБЩАЯ СУММА S='T:END

```

Работу этой программы можно проверить с помощью таблицы, имеющей $M=4$ и $N=5$:

где Int — операция выделения целой части выражения.

y_i	x_i				
	x_1	x_2	x_3	x_4	Σx_i
y_1	25	47	36	54	162
y_2	28	15	10	14	67
y_3	12	74	68	35	189
y_4	87	51	34	14	186
y_5	22	18	11	56	107
Σy_i	174	205	159	173	711

В финансово-экономических расчетах нередко возникает необходимость в определении дня недели по дате числа, а также вычисления количества дней N , прошедших между двумя датами. При отсчете дат с 1582 года эта задача решается с помощью следующего алгоритма (взят из описания библиотеки программ программируемого микрокалькулятора TI-58C).

1. Вводим число (день) D , месяц M и год Y .

3. День недели D_n вычисляется по формуле

$$D_n = F + 7 (-\text{Int}(F/7)),$$

причем $D_n=0$ — суббота, 1 — воскресенье, 2 — понедельник, ..., 6 — пятница.

4. Для нахождения N выполняем п. 1—3 для обеих дат и вычисляем $N = \text{ABS}(F_1 - F_2)$, где ABS — операция выделения абсолютного значения, F_1 — фактор для первого и F_2 — для второго числа.

Программа 7.72.

```

10 PRINT 'ВЫЧИСЛЕНИЕ ПО ДАТЕ ДНЯ НЕДЕЛИ И ЧИСЛА ДНЕЙ МЕЖДУ'
20 PRINT ' ДВУМЯ ДАТАМИ, НАЧИНАЯ С 1582 ГОДА'
30 INPUT 'ВВЕДИТЕ ЧИСЛО, МЕСЯЦ, ГОД 'D,M,Y
50 GOSUB 80:LETT=F
60 INPUT 'ВВЕДИТЕ- ДЕНЬ, МЕСЯЦ, ГОД 'D,M,Y
70 GOSUB 80:PRINT!9.0!'ЧИСЛО ДНЕЙ N='ABS(F-R):STOP
80 LET F=365*Y+D+31*(M-1)
90 IF M<=2 THEN LET R=
100 LET F=F-INT(.4*M+2.3)+INT(Y/4)
110 LET F=F-INT(3/4*(INT(Y/100)+1)):GOTO 130
120 LET F=F+INT((Y-1)/4)-INT(3*(INT((Y-1)/100)+1)/4)
130 LET W=F+(-INT(F/7)*7):PRINT 'ДЕНЬ НЕДЕЛИ'
140 IF W=1 THEN PRINT 'ВОСКРЕСЕНЬЕ'
150 IF W=2 THEN PRINT 'ПОНЕДЕЛЬНИК'
160 IF W=3 THEN PRINT 'ВТОРНИК'
170 IF W=4 THEN PRINT 'СРЕДА'
180 IF W=5 THEN PRINT 'ЧЕТВЕРГ'
190 IF W=6 THEN PRINT 'ПЯТНИЦА'
200 IF W=0 THEN PRINT 'СУББОТА'
210 RETURN:END

```

2. Вычисляем фактор F для января и февраля ($M=1, 2$) по формуле

$$F = 365Y + D + 31(M-1) + \text{Int}((Y-1)/4) - \text{Int}(3/4(\text{Int}(((Y-1)/100)+1)),$$

а для остальных месяцев — по формуле

$$F = 365Y + D + 31(M-1) - \text{Int}(0,4M+2,3) + \text{Int}(Y/4) - \text{Int}(3/4(\text{Int}(Y/100)+1)),$$

Пример А. Для 01.05.86 получим день недели — четверг.

Пример В. День начала Великой Отечественной войны — 22.06.41 (воскресенье), окончания — 09.05.45 (среда), число дней между этими датами 1417.

В [8] описаны алгоритмы для перевода дат из одного вида календаря в другой и примеры вычислений такого рода.

ПОДГОТОВКА К РАБОТЕ СИСТЕМЫ ПОДГОТОВКИ ПРОГРАММ НА БАЗЕ МИКРОЭВМ ЭЛЕКТРОНИКА-ДЗ-28

1. Включение системы в сеть производится с помощью тумблеров, установленных на микроЭВМ, дисплее и принтере. После включения на клавишном пульте дисплея нужно нажать клавиши ДУП, ЛИН и РЕД (см. рис. 1.6).

2. В отсек магнитофона микроЭВМ вставляется кассета с магнитной лентой, содержащей программу-интерпретатор. Нажатием клавиш пульта микроЭВМ Р (работа — установка микроЭВМ в рабочее состояние), С (сброс) и СЛ (считывание с ленты) осуществляется пуск магнитофона и запись программы-интерпретатора в ОЗУ. Если загорается индикатор сбоя записи, нажав клавиши С и СЛ, повторяем запись.

3. Проверим сумму кодов программы-интерпретатора, нажав клавишу КП (контроль программы) на пульте микроЭВМ. Сумма кодов должна соответствовать указанной на кассете с программой-интерпретатором (она индицируется индикатором микроЭВМ).

4. Запускаем программу-интерпретатор, нажав клавишу S (start) микроЭВМ. На экране дисплея появляется запись (начальный диалог):

БЕЙСИК ДЗ-28, ВАРИАНТ 3А
СНИМИТЕ КАССЕТУ

Во избежание ошибочной записи кассета с программой-интерпретатором снимается. Начиная с этого момента управление работой системы производится с пульта дисплея. Нажав клавишу пульта дисплея ПС, получим запрос

В КОМПЛЕКС ВХОДИТ ПЕЧАТАЮЩЕЕ
УСТРОЙСТВО (0 — НЕТ, 1 — ДА)?

На этот запрос надо ввести код 0 или 1. Допустим, ввели код 1 и нажали клавишу ПС. Получим новый запрос:

ВВЕДИТЕ ТИП ПЕЧАТАЮЩЕГО
УСТРОЙСТВА

(0 — ТПУ, 1 — УВВПЧ-30-0004, 2 — CONSUL, 3 — ROBOTRON)? Теперь нужно набрать код подключенного к микроЭВМ печатающего устройства (принтера) и нажать клавишу ПС.

Получаем еще один запрос:

НОМЕРА ВНЕШНИХ ПОДПРОГРАММ?

Если пользователь намерен использовать внешние подпрограммы, нужно указать их номера. В данной книге внешние подпрограммы не используются, поэтому достаточно нажать клавишу ПС еще раз. Начальный диалог заканчивается выдачей надписи.

ГОТОВ

:

5. После двоеточия пользователь может вводить строки программы. По завершении ввода каждой строки нажимается клавиша ПС и на следующей строке появляется знак : (двоеточие). Таким образом, никакой специальной команды для набора программы с пульта не требуется.

6. Если нужно ввести программу с магнитной ленты, кассета с ней вставляется в отсек магнитофона и лента перематывается так, чтобы нужная программа заведомо была в правой части кассеты. На пульте задается команда LOAD 'ИМЯ', где имя — код программы, и нажимается клавиша ПС. МикроЭВМ обеспечивает поиск и считывание нужной программы. Если код (имя) программы не указан, будет считана первая из программ на ленте.

7. Запись программ на магнитную ленту производится командой

SAVE 'ИМЯ'.

8. Если нужно считать или записать не все строки программы, а только от номера НС1 до номера НС2 (или только одну НС), соответствующие номера (разделенные запятой) указываются после команд считывания и записи.

9. Листинг программы можно вывести на экран дисплея, дав команду LIST ≠ 0 (или при первом включении LIST) и нажав клавишу ПС. Команда LIST ≠ 1 обеспечивает распечатку листинга программы принтером. Команды LIST ≠ 0 НС и LIST ≠ 1 НС обеспечивают вывод (и распечатку) одной строки НС, а команды LIST ≠ 0 НС1, НС2 и LIST ≠ 1 НС1, НС2 — вывод (и распечатку) листинга, начиная со строки НС1 и кончая строкой НС2.

10. Пуск программы с начала обеспечивается вводом команды RUN, а со строки НС — вводом команды GOTO НС. После этого нажимается клавиша ПС.

11. Если в ходе ввода и выполнения программы выявляется ошибка, на экран дисплея выводится сообщение

ОШИБКА (номер) В СТРОКЕ (номер НС)

Содержание ошибок и их номера даны в Приложении 2. После устранения ошибки можно начинать вычисления, начиная со строки НС (см. п. 10).

12. Основы программирования, правила редактирования и отладки программ описаны в гл. 2. Отметим, что для распечатки результатов вычислений принтером после команд PRINT должен стоять знак #1 (это указание запоминается и может в дальнейшем не повторяться, если в программе далее не указывался знак вывода данных на экран дисплея #0). Ошибку в строке устраняют ее повторным вводом.

Приложение 2

НОМЕРА ОШИБОК И ИХ СОДЕРЖАНИЕ ДЛЯ СИСТЕМ ПОДГОТОВКИ ПРОГРАММ НА БАЗЕ МИКРОЭВМ ЭЛЕКТРОНИКА ДЗ-28

Номер ошибки	Содержание ошибки
0	Переполнение памяти, отведенной пользователю
1	Недопустимый оператор
2	Переполнение строки ввода
3	Недопустимый ограничитель в строке
4	Недопустимый номер строки НС
5	Несоответствие кавычек в предложении
6	Отсутствие открывающей скобки перед аргументом функции
7	Недопустимый оператор LET
10	Неправильная запись индексов
11	Неправильная размерность индекса, не определен массив
12	Несоответствие скобок в выражении
13	Недопустимый элемент выражения
14	Не определена функция пользователя
15	Неправильное имя переменной
20	Неправильная операция отношения
21	Недопустимый оператор IF
22	Неправильные операторы DIM, COM
23	Недостаточно места для массива
24	Неправильный оператор DEF
25	Нет данных для оператора READ
26	Недопустимый оператор DATA
27	Неправильный формат команд в операторе CMD
30	Неправильный формат оператора FOR — NEXT
31	Нет оператора NEXT
32	Не было FOR
33	Переполнение стека FOR — NEXT
34	Нулевой шаг FOR
35	Неправильный формат PRINT
36	Неправильно задан формат печати
37	Недопустимое выражение в TAB
38	Отсутствие открывающей записи в буфере магнитной ленты
40	Номер следующей строки меньше номера предыдущей
41	Превышение уровня вложения подпрограмм
42	RETURN без GOSUB
43	Нет строки для перехода по GOSUB или GOTO
44	Нет внешней программы с указанным номером
50	Неправильное предложение с операторами обслуживания магнитной ленты и перфоленты
52	Сбой структуры файла
53	Отсутствие в ЗУ массива при приеме магнитной ленты или перфоленты
54	Не считан очередной блок данных с магнитной ленты
55	Считанный с магнитной ленты блок не умещается в ОЗУ
59	При загрузке или записи программы с магнитной ленты указан только один номер строки

Номер ошибки	Содержание ошибки
60	Нет ответа печатающего устройства
61	Нет ответа считывателя
62	Нет ответа перфоратора
121	Недопустимые знаки при вводе по INPUT
122	Недостаточно данных для INPUT
123	Несуществующая переменная
124	Слишком много данных для INPUT
128	Некорректная операция (ОП) в процессе вычисления

Приложение 3

ПОДГОТОВКА ТЭВМ FX-702P К РАБОТЕ

1. Включение ТЭВМ производится переводом выключателя питания в положение ON. Если выключение ТЭВМ осуществлялось автоматически, то нажимается клавиша \overline{AC}_{DN} . На индикаторе появляется надпись READY P0.

2. Для записи программ необходимо перевести ТЭВМ в режим записи, нажав клавиши MODE и 1. В общем случае при этом индицируется надпись вида

WRT	DEG	279
READY P0 : - - - 2 3 * 5 - 7 * 9		

Это означает: режим записи WRT, установка градусной меры измерения углов DEG, установка программной области P0 (READY P0): заняты программные области 0, 1, 6 (на их месте черточки), заняты закодированными программами программные области 4 и 8 (на их месте знак *), свободны программные области 2, 3, 5, 7 и 9, число свободных программных шагов 279 (цифры в правом верхнем углу индикатора). Пользователь может ввести программу в любую из свободных программных областей, если число свободных шагов достаточно. Для этого нужно дать команду F_n P_n, где n — номер свободной области. При этом вместо надписи P0 появится надпись P_n, и можно начать ввод программы. Ввод каждой строки фиксируется нажатием клавиши EXE.

3. Если необходимо стереть какую-либо одну программную область (кроме содержащих закодированные программы), следует ввести команды F₁ P_n CLR EXE. Полное стирание, например, после смены источников питания, выполняется набором следующих команд:

MODE 1 CLR ALL EXE

MODE 0 F₁ SAC F₂ DEFM 0 EXE VAC EXE
ROM 0 EXE AC

Подчеркнутые команды вводятся побуквенно. После выполнения этих команд в режиме MODE 1 на дисплее появляются следующие надписи (см. внизу полосы).

Они означают, что в распоряжении пользователя 26 переменных (variables) и 1680 шагов программы. Если задать после оператора DEFM другое число (до 20), будет указано перераспределение памяти. Например, если дать команду DEFM 5 EXE, получим следующее сообщение VAR : 76 PRG 1280 (т. е. число переменных возрастает до 76 (на $5 \times 10 = 50$), а число шагов программы сокращается до 1280).

4. После ввода программы нужно перейти в режим вычислений, дав команды MODE 0 и P_n. Дальнейшие правила работы задаются заложенными в программу диалогом.

5. Редактирование программ проводится в режиме их записи MODE 1. Вызов любой строки с номером HC задается нажатием клавиш AC (общее стирание дисплея), LIST и HC (номер строки). При команде LIST выводится первая строка (с наименьшим номером). Если операторы строки не помещаются в формате дисплея (20 знакомест), то, нажав клавишу перевода маркера вправо $\overrightarrow{\square}$, можно сдвинуть строку влево. Исправить любой знак в строке можно, расположив маркер под ним и введя нужный

WRT	DEG	1680
READY P0 : 0 1 2 3 4 5 6 7 8 9		

знак. Маркер передвигается с помощью клавиш \leftarrow и \rightarrow . Если держать клавиши нажатыми свыше 2 с, ускоренное перемещение маркера с одной позиции на другую будет происходить автоматически. Команда F, HOME смещает маркер в нулевую позицию. Стереть ошибочный оператор можно, установив маркер в позицию после него и нажав клавишу C. Для включения нового оператора в текст программы в нужное место подводят маркер и нажимают клавиши F, INS столько раз, сколько знакомест требует оператор. После этого вводится сам оператор.

6. Просмотр листинга программ возможен в режиме MODE 0 при подаче команд LIST (просмотр со строки с наименьшим номером) и LIST HC (просмотр со строки с номером HC). При этом дисплей работает в бегущем режиме, т. е. строки автоматически меняются, а если строка не умещается в формате дисплея, она

автоматически продвигается влево. Скорость бега индикации такова, что прочесть строку удается при определенном навыке. Остановку индикации в любой момент можно провести, нажав клавишу STOP (нажатие клавиши CONT возобновляет бегущий режим индикации). В режиме MODE 1 также возможен просмотр листинга построчно (вводом команды LIST или LIST HC). Однако смена строк при этом производится вручную, нажатием клавиши EXE. Пооперационный просмотр трассы программы проводится в режиме TRACE ON, вводимом нажатием клавиш MODE и 2. При этом, нажимая клавиши STOP и END, будем получать индикацию номера каждой строки и листинг каждого предложения (отдельно). Выключить режим трассировки можно, нажав клавиши MODE и 3 (TRACE OFF).

7. Правила записи и считывания программ (с применением кассетного магнитофона), а также использования принтера описаны в § 2.2.

Приложение 4

НОМЕРА ОШИБОК И ИХ СОДЕРЖАНИЕ ДЛЯ ПЭВМ FX-702P

Номер ошибки	Содержание ошибки
1	Переполнение памяти данных или программ
2	Синтаксическая ошибка (неправильная запись операторов, неверное задание формата и т. д.)
3	Некорректная математическая операция, результат выходит за предел 10^{100}
4	Нет строки, к которой задано обращение операторами GOTO и GSB
5	Неверно задается аргумент, заданы несуществующие индексы для одномерных или двумерных массивов
6	Заданная переменная не определена оператором DEFN
7	Ошибки в организации циклов и подпрограмм (нет оператора RET в подпрограмме; есть оператор RET; но нет подпрограммы; число подпрограмм, вложенных друг в друга, больше 10; неверное вложение циклов; число вложенных друг в друга циклов больше 8; оператор NEXT используется без оператора FOR и наоборот)
8	При вызове листинга или стирании закодированной программы не указано ее имя (код)
9	Нет ответа от периферийных устройств (отключены адаптер FA-2 или принтер)

Приложение 5

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ НЕКОТОРЫХ ЧИСЛЕННЫХ МЕТОДОВ ЧАСТНОГО ПРИМЕНЕНИЯ

§ П5.1. Построение полинома по его действительным корням

Пусть многочлен $P_n = \prod_{i=1}^n (x - x_i)$ с действительными корнями x_i нужно представить в виде полинома $P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$, имеющего $a_n = 1$. Для этого ищем коэффициенты полинома в другой форме: $a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$, где $a_0 = 1$. Они

вычисляются по рекуррентным формулам

$$A_n = (-1)^n \prod_{i=1}^n x_i,$$

$$A_{n-l} = \frac{1}{l} \sum_{k=1}^l (-1)^{k+1} A_{n-l+k} \sum_{i=1}^n \left(-\frac{1}{x_i}\right)^k,$$

где $l = 1, 2, 3, \dots, n-1$. Преобразование $A_i = a_i$ в заданную форму производится переиндексацией, т. е. вместо индекса i в цикле вывода A_i на индикацию выводится индекс $N-i$.

```

10 PRINT 'ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТОВ ПОЛИНОМА ПО'
15 PRINT ' ЕГО ДЕЙСТВИТЕЛЬНЫМ КОРНЯМ'
20 INPUT 'ЗАДАЙТЕ СТЕПЕНЬ ПОЛИНОМА N=':N:DIM A(N+1)
25 FOR I=1 TO N:PRINT!2.0!'ВВЕДИТЕ КОРЕНЬ X'I
30 INPUT A(I):NEXT I
40 LETR=1:FOR I=1 TO N:LETR=R*A(I):NEXT I
45 LETE=1:IF N/2-INT(N/2)<0 THEN LETE=-1
50 LETA(N+1)=E*R:FOR L=1 TO N-1
60 LETS=0:FOR K=1 TO L:LETP=0:FOR I=1 TO N
65 LETE=-1/A(I):IF K/2-INT(K/2)=0 THEN LETE=ABS(E)
70 LETP=P+E:NEXT I
75 LETE=-1:IF (K+1)/2-INT((K+1)/2)=0 THEN LETE=1
80 LETS=S+(E*A(N+1-K-L)*P):NEXT K
90 LETA(N+1-L)=S/L:NEXT L
100 FOR I=1 TO N:PRINT!2.0!'A^N-I!F1.9!'='A(I+N)
110 NEXT I:PRINT 'ВЫЧИСЛЕНИЯ ОКОНЧЕНЫ':END

```

Пример. Пусть $N=4$, $x_1=1$, $x_2=1$, $x_3=-1$ и $x_4=-1$. Получаем $a_3=0$, $a_2=-2$, $a_1=0$ и $a_0=1$, т. е. $P(x)=x^4-2x^2+1$.

Пример. Пусть надо решить две системы линейных уравнений:

$$\begin{bmatrix} 4 & 8 & 0 \\ 8 & 8 & 8 \\ 2 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 6 \end{bmatrix} \text{ и } \begin{bmatrix} 4 & 8 & 0 \\ 8 & 8 & 8 \\ 2 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 32 \\ 4 \end{bmatrix}$$

§ П5.2. Обращение матрицы, вычисление определителя и решение систем линейных уравнений с разными векторами свободных членов

Модифицированным методом Гаусса — Жордана матрица A обращается, что дает матрицу A^{-1} . Одновременно вычисляется определитель D матрицы A . Если задана система уравнений $A \cdot X_i = B_i$, то ее неизвестные находятся по формуле $X_i = A^{-1} \cdot B_i$. При этом матрица A^{-1} вычисляется один раз и можно решать ряд систем уравнений с одинаковой матрицей A .

Программа П5.2.

Введя $N=3$ и элементы матрицы A , получим обратную матрицу

$$A^{-1} = \begin{bmatrix} 0,08333 & -0,08333 & 0,666666 \\ 0,08333 & 0,041666 & -0,33333 \\ -0,166666 & 0,16666 & 0,33333 \end{bmatrix}$$

и определитель $D=96$ матрицы A . Введя столбец свободных членов первого уравнения, получим $x_1=4$, $x_2=-1,5$ и $x_3=-2$. Далее, введя столбец свободных членов второго уравнения, получим $x_1=1$, $x_2=1$ и $x_3=2$.

```

05 PRINT 'ОБРАЩЕНИЕ МАТРИЦЫ, ВЫЧИСЛЕНИЕ ОПРЕДЕЛИТЕЛЯ И РЕШЕНИЕ'
10 PRINT 'СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ С РАЗНЫМИ ВЕКТОРАМИ B(I) '
15 PRINT ' МОДИФИЦИРОВАННЫМ МЕТОДОМ ГАУССА-ЖОРДАНА'
20 INPUT 'ВВЕДИТЕ ПОРЯДОК МАТРИЦЫ N=':N:DIM A(N,N),B(N),S(N)
30 FOR I=1 TO N:FOR J=1 TO N:PRINT!2.0!'ВВЕДИТЕ A'I,J'
40 INPUT A(I,J):NEXT J:NEXT I
50 LETD=1:LETR=1:FOR I=1 TO N:FOR J=I TO N
80 IF A(J,I)<>0 THEN 110
90 NEXT J
100 PRINT 'МАТРИЦА ВЫРОЖДЕННАЯ':GOTO 300
110 GOSUB 190:LETA(I,I)=1/A(I,I):GOSUB 210
120 FOR J=1 TO N:IF J=I THEN 140
130 LETB=-A(J,I):LETA(J,I)=0:GOSUB 230
140 NEXT J:LETD=D/A(I,I):NEXT I:FOR I=N TO 1 STEP -1
150 IF A(I,0)=I THEN 180
160 LETR=-R:FOR K=1 TO N:LETB=A(K,I)
170 LETA(K,I)=A(K,A(I,0)):LETA(K,A(I,0))=B:NEXT K
180 NEXT I:GOTO 240
190 FOR K=1 TO N:LETB=A(I,K):LETA(I,K)=A(J,K)
200 LETA(J,K)=B:NEXT K:LETA(I,0)=J:RETURN
210 FOR K=1 TO N:IF K=I THEN 225
220 LETA(I,K)=A(I,I)*A(I,K):NEXT K:RETURN
225 NEXT K:RETURN
230 FOR K=1 TO N:LETA(J,K)=A(J,K)+B*A(I,K):NEXT K:RETURN
240 PRINT 'ЭЛЕМЕНТЫ ОБРАЩЕННОЙ МАТРИЦЫ A^-1(I,J) '
250 FOR I=1 TO N:FOR J=1 TO N
260 PRINT!2.0!'A^-1(I,J)='!F1.9!A(I,J):NEXT J:NEXT I
265 LETD=D*R:PRINT 'ОПРЕДЕЛИТЕЛЬ ИСХОДНОЙ МАТРИЦЫ D=':D
270 FOR I=1 TO N:PRINT!2.0!'ВВЕДИТЕ B'I=':INPUT B(I):NEXT I
275 FOR I=1 TO N:LETS(I)=0:FOR J=1 TO N
280 LETS(I)=S(I)+B(J)*A(I,J):NEXT J
290 PRINT 'КОРЕНЬ X'I!2.0!'I='!F1.9!S(I):NEXT I:GOTO 270
300 END

```

§ П5.3. Решение системы линейных уравнений методом отражения

Метод отражения реализован следующим алгоритмом прямого хода:

$$A = \sum_{i=k}^n a_{ik}^2, \quad B = A + A_{kk}S, \quad S = (\text{sign } A_{kk}) \sqrt{A},$$

$$C = \sum_{j=k}^n A_{ji}A_{jk}, \quad P = (C + A_{ki}S)/B, \quad i = k, \dots, n,$$

$$D = \sum_{j=k}^n B_jA_{jk}, \quad E = (D + SB_k)/B,$$

$$F = A_{ik}, \quad F = F + S, \quad \text{если } i = k,$$

$$A_{ij} = A_{ij} - P_iF_j, \quad j = k, \dots, n,$$

$$B_i = B_i - FE, \quad i = k, \dots, n,$$

$$\text{и } k = n, n-1, \dots, 1, n = N.$$

Неизвестные получаются при обратном ходе:

$$L = \sum_{k+1}^n A_{ki}B_i, \quad x_k = B_k = (B_k - L)/A_{kk},$$

$$k = n, n-1, \dots, 1.$$

Программа П5.3.

```

10 PRINT 'РЕШЕНИЕ СИСТЕМЫ ИЗ N ЛИНЕЙНЫХ УРАВНЕНИЙ'
15 PRINT ' МЕТОДОМ ОТРАЖЕНИЯ'
20 INPUT 'ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N=':N:DIM A(N,N)
30 FOR I=1 TO N:FOR J=1 TO N
40 PRINT!2.0!'ВВЕДИТЕ A'I',J:INPUT A(I,J)
50 IF J=N THEN PRINT'ВВЕДИТЕ B'I:INPUT A(I,0)
60 NEXT J:NEXT I
70 FOR K=1 TO N:LETA=0:FOR I=K TO N
80 LETA=A+A(I,K)^2:NEXT I:LETS=SGN(A(K,K))*SQRT(A)
90 LETB=A+A(K,K)*S:FOR I=K TO N
100 LETC=0:FOR J=K TO N
110 LETC=C+A(J,I)*A(J,K):NEXT J
120 LETA(0,I)=(C+S*A(K,I))/B:NEXT I:LETD=0
130 FOR J=K TO N:LETD=D+A(J,0)*A(J,K):NEXT J
140 LETE=(D+S*A(K,0))/B:FOR I=K TO N
150 LETF=A(I,K):IF I=K THEN LETF=F+S
160 FOR J=K TO N:LETA(I,J)=A(I,J)-A(0,J)*F:NEXT J
170 LETA(I,0)=A(I,0)-F*E:NEXT I:NEXT K
180 FOR K=N TO 1 STEP -1:LETL=0:FOR I=K+1 TO N
190 LETL=L+A(K,I)*A(I,0):NEXT I
200 LETA(K,0)=(A(K,0)-L)/A(K,K):NEXT K
210 PRINT'РЕЗУЛЬТАТЫ РЕШЕНИЯ':FOR I=1 TO N
220 PRINT!2.0!'X'I:IF I.9!'='A(I,0):NEXT I:END
    
```

Пример. Для системы (N=3)

$$3x_1 + 2x_2 + x_3 = 4,$$

$$x_1 + x_2 - x_3 = 1,$$

$$x_1 - 2x_2 + x_3 = 3$$

получим $x_1 = 0,1$, $x_2 = -0,6$ и $x_3 = 1,7$.

§ П5.4. Решение системы линейных уравнений методом простых итераций

При начальных приближениях x_{i0} ($i=1, 2, \dots, N$) вычисляем последовательные приближения по формуле простых итераций [18]

$$x_{i(j+1)} = x_{i(j)} - \frac{1}{a_{ii}} \left(\sum_{k=1}^n a_{ik}x_{k(j)} - b_i \right)$$

до тех пор, пока $x_{i(j+1)} - x_{i(j)} > \epsilon$, где (j) — номер итерации, ϵ — заданная погрешность вычислений. Итерационный процесс сходится, если величина модуля каждого диагонального элемента матрицы A больше суммы модулей остальных элементов. Необходимо задать начальные приближения в виде вектора $[x_{i0}]$.

Программа П5.4.

```

10 PRINT 'РЕШЕНИЕ СИСТЕМЫ ИЗ N ЛИНЕЙНЫХ УРАВНЕНИЙ'
20 PRINT ' МЕТОДОМ ПРОСТЫХ ИТЕРАЦИЙ'
30 INPUT 'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N=':N:DIM A(N,N),B(N),X(N),Z(N)
40 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕНИЯ E=':E
50 FOR I=1 TO N:FOR J=1 TO N
60 PRINT!2.0!'ВВЕДИТЕ A'I',J:INPUT A(I,J)
70 IF J=N THEN PRINT'ВВЕДИТЕ B'I:INPUT B(I)
80 NEXT J:NEXT I
90 LETS=0:FOR I=1 TO N:PRINT'ЗАДАЙТЕ X'I'(0):INPUT Z(I):NEXT I
100 LETK=0:FOR I=1 TO N:LETX(I)=-B(I)
110 FOR J=1 TO N:LETX(I)=X(I)+A(I,J)*Z(J)
120 NEXT J:IF ABS(X(I)/A(I,I))>E THEN LETK=1
130 LETX(I)=Z(I)-X(I)/A(I,I):NEXT I
140 FOR I=1 TO N:LETZ(I)=X(I):NEXT I
150 LETS=S+1:IF K=1 THEN 100
160 PRINT'РЕЗУЛЬТАТЫ РЕШЕНИЯ'
170 FOR I=1 TO N:PRINT!2.0!'X'I'='!F1.9!X(I):NEXT I
180 PRINT!4.0!'ЧИСЛО ИТЕРАЦИЙ S=':S:END
    
```

Пример. Для системы ($N=3$)

$$\begin{aligned} 4x_1 + 0,24x_2 - 0,08x_3 &= 8, \\ 0,09x_1 + 3x_2 - 0,15x_3 &= 9, \\ 0,04x_1 - 0,08x_2 + 4x_3 &= 20, \end{aligned}$$

задав $\varepsilon = E = 1 \cdot 10^{-4}$ и начальные приближения $x_{10} = x_{20} = x_{30} = 1$, получим: $x_1 = 1,9091$, $x_2 = 3,1949$, $x_3 = 5,0448$ (пятый и последующие знаки после запятой отброшены, так как при $\varepsilon = 1 \cdot 10^{-4}$ они недостоверны) и $N_{итер} = 5$ (число итераций).

§ П5.5. Решение системы линейных уравнений методом Зейделя

При методе Зейделя итерационный процесс подобен описанному для метода простых итераций, однако уточненные значения $x_{i(j+1)}$ сразу подставляются в последующие уравнения. Формула итерационного процесса имеет вид [18]

$$x_{i(j+1)} = x_{i(j)} - \frac{1}{a_{ii}} \left(\sum_{k=1}^{i-1} a_{ik} x_{k(j+1)} + \sum_{k=i}^N a_{ik} x_{k(j)} - b_i \right).$$

Условия сходимости те же, что и для метода простых итераций. Обычно метод Зейделя сходится быстрее, чем метод простых итераций, но не исключена и обратная ситуация:

Программа П5.5.

```

10 PRINT 'РЕШЕНИЕ СИСТЕМЫ ИЗ N ЛИНЕЙНЫХ УРАВНЕНИЙ'
20 PRINT 'ИТЕРАЦИОННЫМ МЕТОДОМ ЗЕЙДЕЛЯ'
30 INPUT 'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N:=N: DIM A(N,N), B(N), X(N), Z(N)
40 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ ВЫЧИСЛЕНИЯ E:=E
50 FOR I=1 TO N: FOR J=1 TO N
60 PRINT!2.0!'ВВЕДИТЕ A'I,J' : J: INPUT A(I,J)
70 IF J=N THEN PRINT'ВВЕДИТЕ B'I: INPUT B(I)
80 NEXT J:NEXT I
90 LETS=0:FOR I=1 TO N:PRINT'ЗАДАЙТЕ X'I'(<0>): INPUT Z(I):NEXT I
100 LETK=0:FOR I=1 TO N:LETX(I)=-B(I)
110 FOR J=1 TO N:LETX(I)=X(I)+A(I,J)*Z(J)
120 NEXT J:IF ABS(X(I)/A(I,I))>E THEN LETK=1
130 LETX(I)=Z(I)-X(I)/A(I,I):LETZ(I)=X(I):NEXT I
150 LETS=S+1:IF K=1 THEN 100
160 PRINT'РЕЗУЛЬТАТ РЕШЕНИЯ'
170 FOR I=1 TO N:PRINT!2.0!'X'I'='!F1.9!X(I):NEXT I
180 PRINT!4.0!'ЧИСЛО ИТЕРАЦИЙ S='S:END.
    
```

Пример. Для системы уравнений ($N=3$)

$$\begin{aligned} 10x_1 + x_2 + x_3 &= 12, \\ 2x_1 + 10x_2 + x_3 &= 13, \\ 2x_1 + 2x_2 + 10x_3 &= 14 \end{aligned}$$

при $\varepsilon = 1 \cdot 10^{-4}$ и $x_{10} = x_{20} = x_{30} = 0$ получим $x_1 = 0,9999$, $x_2 = 0,9999$ и $x_3 = 1$ при $N_{итер} = 5$.

§ П5.6. Решение системы линейных уравнений с переопределенной матрицей

Решается система $AX=B$, где

$$A = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix}$$

и $m \geq n$ (т. е. в общем случае число уравнений в системе больше, чем число неизвестных). Система может быть несовместной. Ее решением считается вектор неизвестных, при котором скаляр $(AX-B) \cdot (AX-B)$ принимает наименьшее значение. Решение сводится к решению системы $[A'A] \cdot [X] = [AB]$, где A' — транспонированная матрица A . Для этого используется метод квадратных корней.

Решение системы $AX=B$ (в нашем случае $A \leftarrow A'A$ и $B \leftarrow A'B$) методом квадратных корней реализуется по следующим формулам прямого хода:

$$a_{ii}^{(1)} = \sqrt{a_{ii}}, \quad a_{ij}^{(1)} = a_{ij}/a_{ii}^{(1)},$$

$$f_i^{(1)} = f_i/a_{ii}^{(1)}, \quad j=1, 2, 3, \dots, n,$$

$$a_{ii}^{(2)} = -\sqrt{a_{ii} - \sum_{k=1}^{i-1} a_{ik}^{(1)2}}, \quad i=1, 2, 3, \dots, n,$$

$$a_{ij}^{(2)} = \left(a_{ij} - \sum_{k=1}^{i-1} a_{ik}^{(1)} a_{kj}^{(1)} \right) / a_{ii}^{(2)}, \quad j > i,$$

$$f_i^{(2)} = \left(f_i - \sum_{k=1}^{i-1} f_k^{(1)} a_{ki}^{(1)} \right) / a_{ii}^{(2)}, \quad x_n = f_n^{(1)} / a_{nn}^{(1)}$$

Затем проводится обратный ход:

$$x_i = \left(f_i^{(1)} - \sum_{k=i+1}^n a_{ik}^{(1)} x_k \right) / a_{ii}^{(1)}, \quad i=n, n-1, \dots, 1.$$

При $m=n$ этим методом возможно решение обычных систем линейных уравнений.

Программа П5.6.

Пример. Для системы уравнений ($n=N=3$, $m=M=6$)

$$\begin{aligned} 0,41x_1 - 0,35x_2 + 0,16x_3 &= 0,22, \\ 0,23x_1 + 0,2x_2 + 0,41x_3 &= 0,84, \\ 0,46x_1 - 0,13x_2 + 0,48x_3 &= 0,81, \\ 0,17x_1 + 0,45x_2 - 0,25x_3 &= 0,37, \\ 0,83x_1 + 0,27x_2 - 0,51x_3 &= 0,59, \\ 0,27x_1 + 0,82x_2 - 0,14x_3 &= 0,95 \end{aligned}$$

получим $x_1 = 1$, $x_2 = 1$ и $x_3 = 1$ при $t_c \approx 15$ с.

```

10 PRINT 'РЕШЕНИЕ ПЕРЕОПРЕДЕЛЕННОЙ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ'
15 PRINT ' МЕТОДОМ КВАДРАТНЫХ КОРНЕЙ':PRINT
20 INPUT 'ВВЕДИТЕ ЧИСЛО НЕИЗВЕСТНЫХ N=':N:DIM C(N)
30 INPUT 'ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ M=':M:DIM A(M,N),B(M)
40 FOR I=1 TO M:FOR J=1 TO N
50 PRINT!2.0!'ВВЕДИТЕ A'I','J':INPUT A(I,J)
60 IF J=N THEN PRINT'ВВЕДИТЕ B'I':INPUT B(I)
70 NEXT J:NEXT I
80 FOR J=1 TO N:FOR K=J TO M:LETS=0:FOR I=1 TO M
90 LETS=S+A(I,J)*A(I,K):NEXT I:LETC(K)=S:NEXT K
100 LETC=0:FOR I=1 TO M:LETC=C+A(I,J)*B(I):NEXT I
110 FOR I=J TO N:LETA(I,J)=C(I):NEXT I:LETC(J)=C:NEXT J
120 LETA(1,1)=SQRT(A(1,1)):FOR J=2 TO N
130 LETA(1,J)=A(J,1)/A(1,1):NEXT J:FOR I=2 TO N
140 LETS=0:FOR K=1 TO I-1:LETS=S+A(K,I)*A(K,I):NEXT K
150 LETA(I,I)=SQRT(A(I,I)-S):FOR J=I+1 TO N
160 LETS=0:FOR K=1 TO I-1:LETS=S+A(K,I)*A(K,J):NEXT K
170 LETA(I,J)=(A(J,I)-S)/A(I,I):NEXT J:NEXT I
180 LETC(I)=C(I)/A(I,1):FOR I=2 TO N
190 LETS=0:FOR K=1 TO I-1:LETS=S+A(K,I)*C(K):NEXT K
200 LETC(I)=(C(I)-S)/A(I,1):NEXT I
210 LETC(N)=C(N)/A(N,N):FOR I=N-1 TO 1 STEP -1
220 LETS=0:FOR K=I+1 TO N:LETS=S+A(I,K)*C(K):NEXT K
230 LETC(I)=(C(I)-S)/A(I,1):NEXT I
240 PRINT'РЕШЕНИЕ СИСТЕМЫ':FOR I=1 TO N
250 PRINT!2.0!'X'I'='!F1.9!C(I):NEXT I:END

```

§ П5.7. Приближенное вычисление нормального решения системы линейных уравнений с вырожденной матрицей

Для решения вычисляется параметр

$$\alpha = 0,5 \sqrt{Nl}.$$

За неизвестные принимаются компоненты вектора X_n , получаемого решением системы следующего вида

$$[(A'A + \alpha E)] \cdot [X_n] = [A'B]$$

методом квадратных корней (см. описание к программе П5.6).

Система уравнений $CX=D$, где C — вырожденная матрица, приближенно решается, если предположить, что C_{ij} и d_i заданы с некоторым l -приближением. Тогда эту систему можно свести к решению системы $AX=B$, так что

$$|C_{ij} - a_{ij}| \leq l, \quad |b_i - d_i| \leq l, \quad 0 < l \leq 0,001.$$

Программа П5.7.

```

10 PRINT 'ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ НОРМАЛЬНОГО РЕШЕНИЯ СИСТЕМЫ'
15 PRINT 'ЛИНЕЙНЫХ УРАВНЕНИЙ С ВЫРОЖДЕННОЙ МАТРИЦЕЙ МЕТОДОМ'
20 PRINT ' КВАДРАТНЫХ КОРНЕЙ':PRINT
25 INPUT 'ЗАДАЙТЕ ПАРАМЕТР ПРИБЛИЖЕНИЯ T=':T
30 INPUT 'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N=':N:DIM A(N,N),B(N),C(N)
40 FOR I=1 TO N:FOR J=1 TO N
50 PRINT!2.0!'ВВЕДИТЕ A'I','J':INPUT A(I,J)
60 IF J=N THEN PRINT'ВВЕДИТЕ B'I':INPUT B(I)
70 NEXT J:NEXT I
80 FOR J=1 TO N:FOR K=J TO N:LETS=0:FOR I=1 TO N
90 LETS=S+A(I,J)*A(I,K):NEXT I:LETC(K)=S:NEXT K
100 LETC=0:FOR I=1 TO N:LETC=C+A(I,J)*B(I):NEXT I
110 FOR I=J TO N:LETA(I,J)=C(I):NEXT I:LETC(J)=C:NEXT J
120 LETP=SQRT(*N)/2:FOR I=1 TO N
130 LETA(I,I)=A(I,I)+P:NEXT I:LETA(1,1)=SQRT(A(1,1))
140 FOR J=2 TO N:LETA(1,J)=A(J,1)/A(1,1):NEXT J
150 FOR I=2 TO N:LETS=0:FOR K=1 TO I-1
160 LETS=S+A(K,I)*A(K,I):NEXT K
170 LETA(I,I)=SQRT(A(I,I)-S):FOR J=I+1 TO N
180 LETS=0:FOR K=1 TO I-1:LETS=S+A(K,I)*A(K,J):NEXT K
190 LETA(I,J)=(A(J,I)-S)/A(I,I):NEXT J:NEXT I
200 LETC(I)=C(I)/A(I,1):FOR I=2 TO N
210 LETS=0:FOR K=1 TO I-1:LETS=S+A(K,I)*C(K):NEXT K
220 LETC(I)=(C(I)-S)/A(I,1):NEXT I
230 LETC(N)=C(N)/A(N,N):FOR I=N-1 TO 1 STEP -1
240 LETS=0:FOR K=I+1 TO N:LETS=S+A(I,K)*C(K):NEXT K
250 LETC(I)=(C(I)-S)/A(I,1):NEXT I
260 PRINT'РЕШЕНИЕ СИСТЕМЫ':FOR I=1 TO N
270 PRINT!2.0!'X'I'='!F1.9!C(I):NEXT I:END

```


Пример. При $n=N=3$ и $t=T=1 \cdot 10^{-6}$ решения системы

$$\begin{aligned} 2x_1 - x_2 + \sqrt{2}x_3 &= 5 + 7\sqrt{2}, \\ 3x_1 + 2x_2 - 3x_3 &= -24, \\ 3x_1 + \sqrt{2}x_2 - \frac{15}{7}x_3 &= -12 - 3\sqrt{2} \end{aligned}$$

получаем в виде $x_1=0,979013$, $x_2=-4,226840$ и $x_3=6,160861$. При $t=0$ по этой программе можно решать обычные системы линейных уравнений.

§ П5.8. Решение задачи нелинейных уравнений методом простых итераций

Описание метода дано в § 4.4. Подпрограмма вычисления $F_i(x_i)$ записывается в виде фрагмента основной программы в строках от 50 до 200 без оператора RETURN.

Программа П5.8.

```
10 PRINT 'РЕШЕНИЕ СИСТЕМЫ НЕЛИНЕЙНЫХ УРАВНЕНИЙ'
15 PRINT ' МЕТОДОМ ПРОСТЫХ ИТЕРАЦИЙ'
20 INPUT 'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N=':DIM X(N),F(N)
25 INPUT 'ЗАДАЙТЕ ПОГРЕШНОСТЬ РЕШЕНИЯ E=':E
30 PRINT 'ВВЕДИТЕ НАЧАЛЬНЫЕ ПРИБЛИЖЕНИЯ'
35 FOR I=1 TO N:PRINT!2.0!'X' I='
40 INPUT X(I):NEXT I
45 REM'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(I)=X(I)+F(I)'
50 LET F(1)=X(1)+X(1)^2+X(2)^2-X(1)+.171875
60 LET F(2)=X(2)+X(1)*X(2)-X(2)+.09375
200 LETS=0:FOR I=1 TO N
210 IF ABS(F(I)-X(I))>E THEN LETS=1
220 LET X(I)=F(I):NEXT I
230 IF S=1 THEN 45
240 FOR I=1 TO N:PRINT!2.0!'X' I='!1.9!F(I)
250 NEXT I:END
```

Пример. В тексте программы (строки 50 и 60) задано решение системы уравнений

$$\begin{aligned} F_1 &= x_1^2 + x_2^2 - x_1 + 0,171875 = 0, \\ F_2 &= x_1 x_2 - x_2 + 0,09375 = 0. \end{aligned}$$

Задав $N=2$, $\epsilon=E=1 \cdot 10^{-3}$, $x_{10}=0,171875$ и $x_{20}=0,09375$, получим $x_1=0,2491442164$ и $x_2=0,124686572$ (верны 3 знака после запятой). Время счета контрольного примера около 5 с.

§ П5.9. Вычисление спектра реакции нелинейной системы с аналитически заданной передаточной характеристикой на гармоническое воздействие

Пусть реакция системы на воздействие $x(t)$ описывается нелинейной передаточной характеристикой $y(x)$. Если $x(t)$ — гармоническое воздействие вида

$$x(t) = x_0 + x_m \cos \omega t,$$

то спектр реакции $y(t)$ будет содержать только косинусоидальные члены вследствие четности функций $x(t)$ и $y(t)$. Спектр $y(t)$ вычисляется по следующему алгоритму (см. § 5.1).

1. Задаем постоянную составляющую x_0 и амплитуду x_m воздействия, число интерва-

лов разбивки периода N и число гармоник спектра.

2. Разделив период $x(t)$ T на N частей $\Delta t = T/N$ (берем $T=1$), находим $t_i = \Delta t i$ и $x_i = x_0 + x_m \cos(2\pi i)$ при $i=0, 1, 2, \dots, N$.

3. Обращаясь к подпрограмме вычисления передаточной характеристики $y(x)$, находим значения $y_i = y(x_i)$.

4. Считая изменения $y_i(t)$ на каждом малом отрезке времени Δt линейными, вычисляем постоянную составляющую реакции $A_0 =$

$$\begin{aligned} &= \frac{1}{N} \sum_{i=1}^N (y_{i-1} + y_i) \text{ и коэффициенты } A_k = \\ &= -\frac{\sin Z}{Z^2 N} \sum_{i=1}^N (y_i - y_{i-1}) \sin(Z(2i-1)), \text{ где } Z = \\ &= \pi k/N \text{ и } k=1, 2, \dots, M. \end{aligned}$$

5. Находим коэффициент гармоник $K_r =$

$$= \sqrt{A_2^2 + A_3^2 + \dots + A_M^2} / A_1.$$

Выражения для A_0 и A_k получаются в результате аналитического интегрирования (5.5) при кусочно-линейной аппроксимации $y(t)$.

Программа П5.9.

Пример. Вычисление $y=f(x)$ задается подпрограммой, записываемой со строки 240 и завершаемой операторами RETURN и END. В тексте программы П5.7 задана квадратичная зависимость $y=x^2$. При $N=20$, $M=5$, $x_0=-1$ и $x_m=1$ получим: $A_0=1,5$; $A_1=-1,98360468$, $A_2=0,4837056046$; $A_3=-2,3725 \cdot 10^{-11}$; $A_4=-2,1711 \cdot 10^{-11}$; $A_5=-2,99625 \cdot 10^{-11}$ и $K_r=0,2438820645$. Знак «минус» у A_k указывает на поворот фазы гармоники на угол π рад (или 180°).

§ П5.10. Регрессия для 16 видов парных зависимостей $y(x)$

Если вид эмпирической зависимости $y(x)$, приближающей парную зависимость, заданную совокупностью значений x_i и y_i , не известен заранее, желательно с помощью одной программы провести регрессию для ряда $y(x)$. Приведенная ниже программа обеспечивает запоминание до 255 пар x_i и y_i и выполняет регрессию для одной или нескольких видов зависимостей $y(x)$. Предусмотрена возможность выбора любых $y(x)$ из 16 зависимостей, отмеченных в табл. 5.1.

Порядок работы с программой следующий.

```

10 PRINT 'АНАЛИЗ СПЕКТРА РЕАКЦИИ НЕЛИНЕЙНОЙ СИСТЕМЫ С ЗАДАННОЙ'
20 PRINT 'АНАЛИТИЧЕСКИ ПЕРЕДАТОЧНОЙ ХАРАКТЕРИСТИКОЙ Y(X) ПРИ'
30 PRINT ' ГАРМОНИЧЕСКОМ ВОЗДЕЙСТВИИ X(T)=X0+X1*СOS(W*T) '
40 INPUT 'ЗАДАЙТЕ ЧИСЛО ИНТЕРВАЛОВ РАЗБЕЖЕНИЯ ПЕРИОДА N='N0
50 INPUT 'УКАЖИТЕ ТРЕБУЕМОЕ ЧИСЛО ГАРМОНИК M='M: DIM A(M), F(M)
60 INPUT 'ЗАДАЙТЕ ПОСТОЯННУЮ СОСТАВЛЯЮЩУЮ X0='X0
70 INPUT 'ЗАДАЙТЕ АМПЛИТУДУ ВОЗДЕЙСТВИЯ X1='X1
80 FOR K=0 TO M: LET A(K)=0: NEXT K
90 LET X=X0+X1:60SUB 240: LET Y0=Y: FOR I=1 TO N0
100 LET X=X0+X1*СOS(2*PI*I/N0):60SUB 240: LET Y1=Y
110 FOR K=1 TO M: LET Z=#PI*K/N0: LET D=Z*(2*I-1)
120 LET C=(Y1-Y0)*SIN(Z)/(Z*2/N0): LET A(K)=A(K)+C*SIN(D): NEXT K
130 LET A(0)=A(0)+(Y0+Y1)/2/N0: LET Y0=Y1: NEXT I
140 PRINT 'СПЕКТР Y(T)=N0+СУММА МК*СOS(K*W*T) '
150 PRINT 'ПОСТОЯННАЯ СОСТАВЛЯЮЩАЯ РЕАКЦИИ N0='A(0)
160 PRINT 'АМПЛИТУДЫ ГАРМОНИК РЕАКЦИИ'
170 LET S=0: FOR K=1 TO M: LET S=S+A(K)*A(K)
180 PRINT '3.0!' A'K IF 1.5!'='A(K)
190 NEXT K: IF A(1)=0 THEN 210
200 PRINT 'КОЭФФИЦИЕНТ ГАРМОНИК КГ='SQR(S-A(1)^2)/ABS(A(1))
210 GOTO 70
220 REM 'СО СТРОКИ 240 ЗАПИСЫВАЕТСЯ ПОДПРОГРАММА ВЫЧИСЛЕНИЯ Y(X) '
240 LET Y=X*X: RETURN: END

```

1. Вводятся значения x_i , y_i и C . Постоянная C задается равной 1, если ввод x_i и y_i будет продолжен. При вводе последней пары x_i и y_i надо задать $C=0$, что обеспечит автоматический переход к исполнению следующей части программы.

2. Регрессия с заданной зависимостью $y(x)$ проводится после указания кода k в виде номера соответствующей зависимости. При задании $k=0$ на индикацию выводится перечень кодов и формул для соответствующих $y(x)$. Результаты регрессии являются ее параметры A и B ($A=b_0$, $B=b_1$, где b_0 и b_1 — обозначения этих параметров в табл. 5.1), а также коэффициент парной корреляции R . Регрессия прово-

дится на основе линеаризующих преобразований, сводящих нелинейную зависимость $y(x)$ к линейной $y=W+Vx$ с последующим преобразованием W и V в A и B .

3. После проведения регрессии задается значение x и код ($C=1$, если будут вычисляться значения $y(x)$ по заданным x , и $C=0$, если в этом нет нужды). В последнем случае вычисления повторяются с п. 2.

Для выбора нужных преобразований и зависимостей $y(x)$ применяется оператор $ON(a)$, обеспечивающий безусловный переход на строку с номером a , где a — арифметическое выражение, значение которого задается кодом k

Программа П5.10.

```

10 PRINT 'РЕГРЕССИЯ ДЛЯ РАЗЛИЧНЫХ ПАРНЫХ ЗАВИСИМОСТЕЙ Y=F(X) '
20 LET M=255: DIM X(M), Y(M)
30 PRINT 'ВВЕДИТЕ X, Y, C (C=1 ПРИ ВВОДЕ, C=0 ПО ОКОНЧАНИИ ВВОДА) '
40 LET J=1: INPUT X(J), Y(J), C: IF C<>0 THEN 30
50 PRINT 'ЗАДАЙТЕ КОД К НУЖНОГО ВИДА РЕГРЕССИИ, K=0, ЕСЛИ НУЖНО '
60 INPUT ' ВМЗВАТЬ ТАБЛИЦУ КОДОВ И Y=F(X) K='K
65 LET X1=0: LET X2=0: LET Y1=0: LET Y2=0: LET R=0: IF K=0 THEN 900
70 IF K=16 THEN INPUT 'ВВЕДИТЕ N='N
75 FOR I=1 TO J: LET X(I)=X: LET Y(I)=Y: ON(200+10*K)
80 LET A=W: LET B=V: ON(500+10*K)
90 PRINT '1.9!' ПАРАМЕТРЫ A='A' B='B'
95 PRINT 'КОЭФФИЦИЕНТ КОРРЕЛЯЦИИ R='R
100 INPUT 'ВВЕДИТЕ X, C(C=1 ИЛИ 0 ПРИ ОКОНЧАНИИ) 'X, C
110 IF C=0 THEN 50
120 ON(700+10*K)
130 PRINT '1.9!' ЗНАЧЕНИЕ Y(X)='Y: GOTO 100
200 REM 'ЛИНЕАРИЗУЮЩИЕ ПРЕОБРАЗОВАНИЯ'
210 GOTO 400
220 LET Y=1/Y: GOTO 400
230 LET X=1/X: GOTO 400
240 LET Y=X/Y: GOTO 400
250 LET Y=LGT(Y): GOTO 400
260 LET Y=LOG(Y): GOTO 400
270 LET Y=LGT(Y): GOTO 400
280 LET X=EXP(-X): LET Y=1/Y: GOTO 400
290 LET X=LGT(X): LET Y=LGT(Y): GOTO 400
300 LET X=LGT(X): GOTO 400
310 LET X=LOG(X): GOTO 400
320 LET Y=1/Y: GOTO 400

```

```

330 LETX=1/X:LETY=1/Y:GOTO 400
340 LETX=1/X:LETY=LOG(Y):GOTO 400
350 LETX=1/X:LETY=LGT(Y):GOTO 400
360 LETX=X^N:GOTO 400
390 REM' ЛИНЕЙНАЯ РЕГРЕССИЯ'
400 LETX1=X1+X:LETY1=Y1+Y:LETX2=X2+X*X
410 LETY2=Y2+Y*Y:LETR=P+X*Y:NEXT I
420 LETU=(X1*Y1-J*P)/(X1*X1-J*X2)
430 LETW=(Y1-U*X1)/J
440 LETR=(P-X1*Y1/J)/SQR((X2-X1*X1/J)*(Y2-Y1*Y1/J))
450 GOTO 80

500 REM' СВЕДЕНИЕ ЛИНЕЙНОЙ РЕГРЕССИИ К НЕЛНЕЙНОЙ'
510 GOTO 90
520 GOTO 90
530 GOTO 90
540 GOTO 90
550 LETA=10^N:LETB=10^U:GOTO 90
560 LETA=EXP(W):GOTO 90
570 LETA=10^W:GOTO 90
580 GOTO 90
590 GOTO 570
600 GOTO 90
610 GOTO 90
620 LETA=1/U:LETB=W/U:GOTO 90
630 LETA=1/W:LETB=U/W:GOTO 90
640 GOTO 560
650 GOTO 570
660 GOTO 90

700 REM' ВЫЧИСЛЕНИЕ Y(X) ПО ЗАДАННОМУ X'
710 LETY=A+B*X:GOTO 130
720 LETY=1/(A+B*X):GOTO 130
730 LETY=A+B/X:GOTO 130
740 LETY=X/(A+B*X):GOTO 130
750 LETY=A*B^X:GOTO 130
760 LETY=A*EXP(B*X):GOTO 130
770 LETY=A*10^(U*X):GOTO 130
780 LETY=1/(A+B*EXP(-X)):GOTO 130
790 LETY=A*X^B:GOTO 130
800 LETY=A+B*LGT(X):GOTO 130
810 LETY=A+B*LOG(X):GOTO 130
820 LETY=A/(B+X):GOTO 130
830 LETY=A*X/(B+X):GOTO 130
840 LETY=A*EXP(B/X):GOTO 130
850 LETY=A*10^(B/X):GOTO 130
860 LETY=A+B*X^N:GOTO 130
900 PRINT' ТАБЛИЦА КОДОВ И ЭМПИРИЧЕСКИХ ФУНКЦИЙ Y(X)'
910 PRINT' 1. Y=A+B*X          2. Y=1/(A+B*X)'
920 PRINT' 3. Y=A+B/X          4. Y=X/(A+B*X)'
930 PRINT' 5. Y=A*B^X          6. Y=A*EXP(B*X)'
940 PRINT' 7. Y=A*10^(B*X)     8. Y=1/(A+B*EXP(-X))'
950 PRINT' 9. Y=A*X^B          10. Y=A+B*LGT(X)'
960 PRINT' 11. Y=A+B*LOG(X)   12. Y=A/(B+X)'
970 PRINT' 13. Y=A*X/(B+X)    14. Y=A*EXP(B/X)'
980 PRINT' 15. Y=A*10^(B/X)   16. Y=A+B*X^N'
990 INPUT' ЗАДАЙТЕ КОД РЕГРЕССИИ K='K
1000 GOTO 65:END

```

Контрольные примеры (везде $x_1=1$, $x_2=2$, $x_3=3,5$, $x_4=5$, номер совпадает с кодом k , а результат дан в скобках).

1. $y_1=0$, $y_2=0,9$, $y_3=2,5$, $y_4=4,2$ ($A=1,13537$, $B=1,0558$, $R=0,99908$, $y(2)=0,97619$, $y(4)=3,0878$).

2. $y_1=0,1$, $y_2=0,05$, $y_3=0,027$, $y_4=0,02$ ($A=0,11086$, $B=10,13857$, $R=0,99845$, $y(3)=0,032758$, $y(1,5)=0,06528$).

3. $y_1=10,3$, $y_2=5$, $y_3=3$, $y_4=2,4$ ($A=0,20763$, $B=10,0062$, $R=0,99880$, $y(1)=10,21384$, $y(3)=3,543034$).

4. $y_1=0,17$, $y_2=0,18$, $y_3=0,19$, $y_4=0,192$ ($A=0,93817$, $B=5,01769$, $R=0,99993$, $y(1)=0,16790$, $y(3)=0,18760$).

5. $y_1=2$, $y_2=4$, $y_3=11$, $y_4=31$ ($A=1,0117$, $B=1,98157$, $R=0,99999$, $y(2)=3,97262$, $y(4)=15,59902$).

6. $y_1=7,4$, $y_2=53$, $y_3=1040$, $y_4=20300$ ($A=1,0168$, $B=1,9800$, $R=0,999999$, $y(2)=53,34312$, $y(4)=2798,46803$).

7. $y_1=3$, $y_2=5$, $y_3=10$, $y_4=20$ ($A=1,90498$, $B=0,20501$, $R=0,99979$, $y(1)=3,05418$, $y(3)=7,85066$).

8. $y_1=2,1$, $y_2=4,2$, $y_3=7,7$, $y_4=9,3$ ($A=0,099793$, $B=1,02288$, $R=0,099999$, $y(2)=4,19771$, $y(4)=8,43686$).

9. $y_1=1$, $y_2=4$, $y_3=12$, $y_4=25$ ($A=0,99935$, $B=1,99493$, $R=0,99998$, $y(2)=3,98338$, $y(4)=15,87764$).

10. $y_1=1, y_2=1.6, y_3=2.1, y_4=2.4$ ($A=0.99924, B=2.00965, R=0.99996, y(2)=1.60421, y(4)=2.20917$).

11. $y_1=1, y_2=2.4, y_3=3.5, y_4=4.2$ ($A=1.00831, B=1.98764, R=0.99997, y(2)=2.38604, y(4)=3.76377$).

12. $y_1=3.3, y_2=2.5, y_3=1.8, y_4=1.4$ ($A=9.70274, B=1.91056, R=0.99986, y(2)=2.48116, y(4)=1.64159$).

13. $y_1=3.33, y_2=5, y_3=6.36, y_4=7.14$ ($A=10.00373, B=2.00375, R=0.999999, y(2)=4.99719, y(4)=6.66499$).

14. $y_1=74, y_2=27, y_3=17, y_4=15$ ($A=9.80145, B=2.01995, R=0.99963, y(2)=26.91020, y(4)=16.24066$).

15. $y_1=500, y_2=50, y_3=19, y_4=13$ ($A=5.16152, B=1.98393, R=0.99998, y(2)=50.66928, y(4)=16.1719$).

16. $y_1=2.1, y_2=2.28, y_3=2.65, y_4=3.12$ ($A=1.99742, B=0.10022, R=0.99998, y(2)=2.28001, y(4)=2.79915$).

В этих примерах результаты указаны с точностью до 5–6 знаков после запятой. Время выполнения каждого контрольного примера составляет несколько секунд (при четырех парах x_i, y_i). С ростом числа пар x_i, y_i оно пропорционально растет.

§ П5.11. Сплайн-аппроксимация, интерполяция и экстраполяция

Полносиманальная интерполяция и аппроксимация не обеспечивает непрерывность производных функции $y(x)$ и может давать значительные погрешности в промежутках между узлами. Кроме того, она плохо приспособлена для экстраполяции и, как правило, не обеспечивает правильного асимптотического поведение $y(x)$ при изменении аргумента x за пределами интервала интерполяции. Нередко с увеличением числа узлов погрешность такой интерполяции не только не уменьшается, но и начинает расти.

От этих недостатков свободна аппроксимация и интерполяция с помощью сплайн-функций. Сплайн (spline) в переводе с английского языка означает «гибкая линейка». Сплайн-функцию можно наглядно трактовать как линию, которую образует гибкая линейка, будучи закрепленной в ряде точек — узлах интерполяции. Математически сплайн — специальный многочлен, принимающий в узлах значения $y(x) = y_i = y(x_i)$ и обеспечивающий непрерывность в них производных. Обычно достаточно обеспечить непрерывность первой и второй производных, для чего

достаточно использовать сплайн-многочлены третьего порядка (кубические сплайны) [18, 24, 42].

Для каждого отрезка $[x_i, x_{i+1}]$ изменения x кубическая сплайн-функция записывается в виде

$$f_i(x) = \frac{1}{6h_i} [m_i(x_{i+1}-x)^3 + m_{i+1}(x-x_i)^3] + \frac{1}{h_i} \left[\left(y_i - \frac{m_i h_i^2}{6} \right) (x_{i+1}-x) + \left(y_{i+1} - \frac{m_{i+1} h_i^2}{6} \right) (x-x_i) \right],$$

где $h_i = x_{i+1} - x_i$, $f_i(x) = y(x)$, $m_i = f''(x_i)$ и $i = 1, 2, \dots, n$ (n — число узлов). При известных x_i, y_i и m_i эта формула задает сплайн-аппроксимацию.

Если потребовать выполнения условия $f_i(x) = y_i$, то приведенное выше выражение для кубических полиномов-сплайнов приводит к системе линейных уравнений, из которых находятся m_i :

$$h_i m_i + 2(h_i + h_{i+1}) m_{i+1} + h_{i+1} m_{i+2} = 6 \left(\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \right).$$

Однако эта система не полностью определяет m_i . Чтобы полностью определить все m_i , нужно задать дополнительные граничные условия. Если они заданы в виде $m_1 = 0$ и $m_n = 0$, получаем нормальные сплайн-функции; при $m_n = m_1$ и $m_{n+1} = m_2$ имеем периодические сплайн-функции и т. д.

Приведенная ниже программа для $n = N$ значений x_i и y_i вычисляет по данному алгоритму коэффициенты $m_i = f''(x_i)$ нормального кубического сплайна и при $x \in [x_1, x_n]$ обеспечивает интерполяцию с помощью описанной выше сплайн-функции. Если $x < x_1$, то выполняется линейная экстраполяция по формуле

$$f(x) = y_1 - ((x_2 - x_1) m_2 / 6 + (y_2 - y_1) / (x_2 - x_1)) (x - a_1),$$

а при $x > x_n$ — экстраполяция по формуле

$$f(x) = y_n + ((x_n - x_{n-1}) m_{n-1} / 6 + (y_n - y_{n-1}) / (x_n - x_{n-1})) (x - x_n).$$

При экстраполяции предполагается, что наклон линейного участка $f(x)$ равен первой производной сплайн-функции в точке (x_1, y_1) при экстраполяции для $x < x_1$ и в точке (x_n, y_n) при $x > x_n$.

Программа П5.11.

```

010 PRINT 'СПЛАЙН-АППРОКСИМАЦИЯ, ИНТЕРПОЛЯЦИЯ И ЭКСТРАПОЛЯЦИЯ'
020 INPUT 'ЗАДАЙТЕ ЧИСЛО УЗЛОВ N=' N
030 DIM X(N), Y(N), L(N), M(N), R(N), S(N)
040 FOR I=1 TO N:PRINT I;2.0:FOR I=1
050 INPUT 'ВВЕДИТЕ X,Y,X(I),Y(I):NEXT I
060 LET D=X(2)-X(1):LETE=(Y(2)-Y(1))/D
070 FOR K=2 TO N-1:LETH=D:LETD=X(K+1)-X(K)
080 LET F=E:LETE=(Y(K+1)-Y(K))/D:LETL(K)=D/(D+H)
090 LETR(K)=1-L(K):LETS(K)=6*(E-F)/(H+D):NEXT K
100 FOR K=2 TO N-1:LETP=1/(R(K)*L(K-1)+2):LETL(K)=-L(K)*F
110 LETS(K)=(S(K)-R(K))*S(K-1)*F:NEXT K
120 LETM(N)=B:LETL(N-1)=S(N-1):LETM(N-1)=L(N-1)
130 FOR K=N-2 TO 1 STEP-1
140 LETL(K)=L(K)*L(K+1)+S(K):LETM(K)=L(K):NEXT K

```

```

150 FOR K=1 TO 2:PRINT 2.0!*M(K).9!*M(K):NEXT K
160 INPUT 'ВВЕДИТЕ АРГУМЕНТ X='X:LET I=0:IF X<X(N) THEN 230
170 IF X<X(1) THEN 250
180 LET I=I+1:IF X>X(I) THEN 180
190 LET J=I-1:LETD=X(I)-X(J):LETH=X-X(J):LETR=X(I)-X
200 LET P=D*D/6:LET Y=(M(J)*R^3+M(I)*H^3)/6/D
210 LET V=Y+(Y(J)-M(J)*P)*R+(Y(I)-M(I)*P)*H)/D
220 PRINT 'ЗНАЧЕНИЕ Y(X)='Y:GOTO 160
230 LET D=X(N)-X(N-1):LET Y=D*M(N-1)/6+(Y(N)-Y(N-1))/D
240 LET V=Y*(X-X(N))+Y(N):GOTO 220
250 LET D=X(2)-X(1):LET Y=-D*M(2)/6+(Y(2)-Y(1))/D
260 LET V=Y*(X-X(1))+Y(1):GOTO 220:END

```

Контрольный пример. Пусть надо провести сплайн-аппроксимацию для интеграла Френеля $S(x)$, заданного таблицей из $N=7$ пар x_i и y_i :

$x_i=0$	0,3	0,5	0,8	1	1,3	1,5
$y_i=0$	0,299401	0,4923442	0,7228442	0,7798934	0,6385505	0,4452612

Введя N , x_i и y_i , получим: $m_1=0$, $m_2=-0,08877862525$, $m_3=-0,5547268737$, $m_4=-2,019378004$, $m_5=-3,563639669$, $m_6=-1,902729099$, $m_7=0$. Далее, задавая x , можно провести интерполяцию и экстраполяцию: $y(0,2)=0,2000938813$ (0,1999211), $y(0,48)=-0,4737408625$ (0,473751), $y(1,6)=0,3422741197$ и т. д. (в скобках после результата даны точные значения).

инвертирование матрицы 3×3 (если индексация массивов идет с 1) или 4×4 (если индексация идет с 0).

```

10 REM 'ТРАНСПОНИРОВАНИЕ МАТРИЦЫ A'
20 INPUT 'ЗАДАЙТЕ N='N: N=N-1: DIM A(N,N): B(N,N)
30 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9
40 MAT READ A: MAT B=TRN(A)
50 PRINT '!F.1.5!' МАССИВ ТРАНСПОНИРОВАННОЙ МАТРИЦЫ ПОСТРОЧНО
60 MAT PRINT B: END

```

В приведенных ниже программах индексация массивов идет с 0, операторы DIM задают точные размеры матриц (с учетом принятой индексации), в операторах MAT матрицы задаются лишь именами массивов. Даны ссылки на программы, обеспечивающие те же вычисления без использования матричных операторов и сравнение программ.

Программа П5.12/1.

Хотя сплайн-интерполяция осуществима при переменном шаге $h_i = x_{i+1} - x_i$, значительное изменение h_i ($h_{i \max}/h_{i \min} > 4$) нецелесообразно, поскольку в этом случае преимущества сплайн-интерполяции теряются и погрешность ее заметно возрастает. С применением сплайн-интерполяции возможно численное интегрирование, спектральный анализ и другие математические операции.

§ П5.12. Пакет программ с матричными операторами

Применение матричных операторов (см. § 2.2) сокращает текст программ, содержащих операции с матрицами, и позволяет в 5—20 раз (в зависимости от типа ПЭВМ) уменьшить время вычислений. Сокращение времени вычислений обусловлено тем, что основные матричные операции при этом выполняются на машинно-ориентированном языке ЭВМ.

Перед использованием матричных операторов с помощью операторов DIM задаются размеры массивов (основных и служебных). Далее с помощью операторов DATA, MAT READ, INPUT или MAT INPUT вводятся элементы исходных матриц. В некоторых версиях бейсика размеры матриц до 10×10 можно не задавать, а размеры больших матриц задавать с запасом. При этом точные размеры матриц указываются в операторе, например, MAT INV A (3, 3) означает

Пр и м е р. Эта программа обеспечивает транспонирование матрицы, взятой из примера к программе 3.48. После задания размерности N квадратной матрицы A , уменьшения N на 1 и задания размерности массивов $A(N, N)$ и $B(N, N)$ в строке 20 с помощью оператора DATA (строка 30) построчно вводятся элементы исходной матрицы A . В строке 40 оператор MAT READ A обеспечивает формирование матрицы A , а оператор MAT B=TRN(A) — транспонирование матрицы A . После комментария в строке 50 массив транспонированной матрицы B выводится построчно на индикацию с помощью оператора MAT PRINT B (строка 60). Отметим, что в состав последнего в данной версии бейсика нельзя включать знаки изменения формата. Поэтому эти знаки (!F.1.5!) даны после оператора PRINT в предшествующей строке 50. Результат получается в виде чисел 1, 4, 7, 2, 5, 8, 3, 6 и 9 (соответствует матрице A^T для контрольного примера к программе 3.48).

Программа П5.12/2.

Эта программа обеспечивает инвертирование матрицы A . Она примерно в 5 раз короче программы 3.56 и обеспечивает уменьшение времени обращения примерно в 20 раз. Значения элементов исходной матрицы и результаты вычислений соответствуют результатам, приведенным для контрольного примера к программе 3.55.

```

10 REM'ОБРАЩЕНИЕ МАТРИЦЫ A'
20 INPUT'ЗАДАЙТЕ N=N: N=N-1: DIM A(N,N),B(N,N),C(N,N)
30 DATA 4,8,0,8,8,8,2,0,1
40 MAT READ A: MAT A=INV(B,C)
50 PRINT!F1.5!'ОПРЕДЕЛИТЕЛЬ МАТРИЦЫ A D='B(0)
60 PRINT'МАССИВ ОБРАЩЕННОЙ МАТРИЦЫ ПОСТРОЧНО'
70 MAT PRINT A:END
    
```

Программа П5.12/3.

```

10 REM'УМНОЖЕНИЕ МАТРИЦ'
20 DIM A(3,4),B(4,2),C(3,2)
30 DATA 1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9,1,2
40 MAT READ A
50 DATA 1,2,3,1,2,3,1,2,3,1,2,3,1,2,3
60 MAT READ B
70 MAT C=A*B
80 PRINT !F1.5!'ВЫВОД ПОСТРОЧНО ЭЛЕМЕНТОВ МАТРИЦЫ C'
90 MAT PRINT C:END
    
```

Программа иллюстрирует умножение двух матриц A и B с применением оператора $MAT C = A * B$. Числовые значения элементов матриц A и B соответствуют контрольному примеру к программе 3.58. Следует отметить, что в этом примере размеры всех матриц различны. Элементы результата — матрицы C — выводятся построчно: 15, 30, 45, 31, 62, 93, 20, 40, 60, 27, 54 и 81.

Программа П5.12/4.

```

10 PRINT'ОБРАЩЕНИЕ МАТРИЦЫ A, ВЫЧИСЛЕНИЕ ЕЕ ОПРЕДЕЛИТЕЛЯ'
20 PRINT'И РЕШЕНИЕ СИСТЕМЫ УРАВНЕНИЙ A*X=B ПРИ РАЗНЫХ B'
30 INPUT'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N=N: N=N-1
40 DIM A(N,N),B(N),C(N,N),D(N,N),X(N)
50 FOR I=0 TO N:FOR J=0 TO N:PRINT!2.0!'ВВЕДИТЕ A'I+1','J+1
60 INPUT A(I,J):NEXT J:NEXT I:MAT A=INV(C,D)
70 PRINT!F1.5!'ВЫВОД ЭЛЕМЕНТОВ ОБРАЩЕННОЙ МАТРИЦЫ'
80 MAT PRINT A:PRINT'ОПРЕДЕЛИТЕЛЬ МАТРИЦЫ A D='C(0)
90 INPUT'БУДЕТ ЛИ РЕШАТЬСЯ СИСТЕМА УРАВНЕНИЙ? ДА 1,НЕТ 0 'K
100 IF K=0 THEN PRINT'ВЫЧИСЛЕНИЯ ОКОНЧЕНЫ':END
110 FOR I=0 TO N:PRINT!2.0!'ВВЕДИТЕ B'I+1
120 INPUT B(I):NEXT I:MAT X=A*B
130 PRINT!F1.5!'МАССИВ НЕИЗВЕСТНЫХ СИСТЕМЫ'
140 MAT PRINT X:PRINT'КОНЕЦ':GOTO 90:END
    
```

Эта программа обеспечивает ввод с пульта элементов квадратной матрицы A размером $N \times N$, обращение матрицы A , вычисление ее определителя, ввод вектора с числом компонентов N и решение системы линейных уравнений $AX=B$ для разных B методом обращения матрицы, т. е. вычисление $X=A^{-1}B$. Программа аналогична программе П5.2, но вдвое короче ее и обеспечивает уменьшение времени вычисления примерно

в 10 раз. Программу П5.12/4 можно проверить по контрольному примеру, приведенному к программе П5.2.

Решение системы уравнений $AX=B$ с применением формулы $X=A^{-1}B$ практикуется редко, так как другие методы (например, Гаусса) обеспечивают в 2—3 раза меньшее время вычислений — при условии одинаковой реализации. Однако резкое уменьшение времени обращения матриц с помощью специального матричного

оператора $MAT INV$ и оператора умножения матриц в конечном счете обеспечивает существенное уменьшение времени вычислений. Этот пример наглядно показывает, что наличие специальных операторов в некоторых версиях бейсика может существенно влиять на целесообразность применения того или иного численного метода.

Программа П5.12/5.

```

10 PRINT'РЕШЕНИЕ СИСТЕМЫ НЕЛИНЕЙНЫХ УРАВНЕНИЙ'
20 PRINT'МОДИФИЦИРОВАННЫМ МЕТОДОМ НЬЮТОНА.ВАР.2'
30 INPUT'ЗАДАЙТЕ ЧИСЛО УРАВНЕНИЙ N=N: N=N-1
40 DIM X(N),B(N),F(N),A(N,N),K(N,N),L(N,N)
50 INPUT'ЗАДАЙТЕ МАКСИМАЛЬНОЕ ЧИСЛО ИТЕРАЦИЙ M=M
60 INPUT'ЗАДАЙТЕ ОТНОСИТЕЛЬНУЮ ПОГРЕШНОСТЬ E=E: S=0
70 FOR I=0 TO N:PRINT!2.0!'ВВЕДИТЕ X'I(0)':INPUT X(I):NEXT I
80 GOSUB 180 : MAT B=F
90 FOR J=0 TO N: Y=X(J): H=E*ABS(Y)
100 X(J)=Y+H:GOSUB 180:FOR I=0 TO N
110 A(I,J)=(F(I)-B(I))/H:NEXT I: X(J)=Y:NEXT J
120 S=S+1:IF S=M+1 THEN PRINT'ЧИСЛО ИТЕРАЦИЙ S=M':END
130 MAT A=INV(K,L):MAT B=A*F:MAT X=X-B
140 R=0:FOR I=0 TO N:IF ABS(B(I)/X(I))>E THEN R=1
    
```

```

150 NEXT I: IF R=1 THEN GO
160 PRINT!F1.9!'МАССИВ НЕИЗВЕСТНЫХ СИСТЕМ'
170 MAT PRINT X:PRINT!2.0!'ЧИСЛО ИТЕРАЦИЙ S='S:END
180 REM'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F(I)=F(X(0),X(1),...,X(N-1))'
190 F(0)=X(0)+3*LG(X(0))-X(1)*X(1)
200 F(1)=2*X(0)*X(0)-X(0)*X(1)-5*X(0)+1
210 RETURN:END

```

Эта программа обеспечивает решение систем нелинейных уравнений модифицированным методом Ньютона (Ньютона — Рафсона). Программа аналогична программе 4.20 и может быть проверена по контрольному примеру к последней.

Программа П5.12/6.

```

10 PRINT'ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ И ИНТЕРПОЛЯЦИЯ'
20 INPUT'ЗАДАЙТЕ ЧИСЛО ОТСЧЕТОВ N='N: M=N-1
30 DIM A(M,M),C(M,M),D(M,M),B(M),X(M)
40 FOR I=0 TO M:PRINT!2.0!'ВВЕДИТЕ X'I',Y'I
50 INPUT Z,B(I): A(I,0)=1: R=1:FOR J=1 TO M
60 R=R*Z: A(I,J)=R:NEXT J:NEXT I
70 MAT A=INV(C,D):MAT X=A*B
80 PRINT!F1.9!'КОЭФФИЦИЕНТЫ A0,A1,...,AM'MAT PRINT X
90 INPUT'ВВЕДИТЕ X='Z: Y=Z*X(M)
100 FOR I=M-1 TO 1 STEP-1: Y=(Y+X(I))*Z:NEXT I
110 Y=Y+X(0):PRINT'Y(X)='Y:GOTO 90:END

```

Программа П5.12/6 обеспечивает аппроксимацию таблицно заданной n значениями x_i и y_i функции $y_i(x_i)$ с помощью полинома

$$y = a_0 + a_1x + a_2x^2 + \dots + a_mx^m,$$

где $m = n - 1$. Вычисление коэффициентов этого полинома осуществляется решением системы линейных (относительно коэффициентов a_0, a_1, \dots, a_m) уравнений

$$a_0 + a_1x_0 + a_2x_0^2 + \dots + a_mx_0^m = y_0,$$

$$a_0 + a_1x_1 + a_2x_1^2 + \dots + a_mx_1^m = y_1,$$

$$a_0 + a_1x_m + a_2x_m^2 + \dots + a_mx_m^m = y_m.$$

Программа позволяет также находить $y(x)$ по заданному x , т. е. обеспечивает интерполяцию.

Данный метод обычно сложнее и требует больших затрат времени, чем аппроксимация и интерполяция методами Ньютона или Лагранжа (§ 4.2). Однако из-за высокой скорости решения систем линейных уравнений с помощью матричных операторов программа П5.12/6 обеспечивает в несколько раз меньше время вычисления коэффициентов a_0, a_1, \dots, a_m и имеет вдвое меньшую длину листинга, чем программа 4.7. Программу П5.12/6 можно проверить по контрольному примеру к программе 4.7.

Программа П5.12/7.

```

10 PRINT'ПОЛИНОМИАЛЬНАЯ АППРОКСИМАЦИЯ И ИНТЕРПОЛЯЦИЯ'
20 PRINT' МЕТОДОМ НАИМЕНЬШИХ КВАДРАТОВ'
30 INPUT'ВВЕДИТЕ СТЕПЕНЬ ПОЛИНОМА M='N: P=2*(N+1)
40 DIM A(N,N),B(N),C(P),D(N,N),E(N,N),X(N)
50 INPUT'ЗАДАЙТЕ ЧИСЛО НАР X,Y N='N
60 FOR I=0 TO N-1:PRINT!3.0!'ДЛЯ I='I:INPUT'ВВЕДИТЕ X,Y 'X,Y
70 F=1:FOR J=0 TO P:IF J>N THEN 90
80 B(J)=E(J)+Y: Y=Y*X
90 C(J)=C(J)+F: F=F*X:NEXT J:NEXT I
100 FOR I=0 TO N: K=I:FOR J=0 TO N
110 A(I,J)=C(K): K=K+1:NEXT J:NEXT I
120 MAT A=INV(D,E):MAT X=A*B
130 PRINT!F1.9!'КОЭФФИЦИЕНТЫ ПОЛИНОМА'MAT PRINT X
140 PRINT:INPUT'ВВЕДИТЕ X='Z: S=0:FOR I=N TO 1 STEP -1
150 S=(S+X(I))*Z:NEXT I: S=S+X(0)
160 PRINT'ЗНАЧЕНИЕ Y(X)='S:GOTO 140:END

```

Программа обеспечивает полиномиальную аппроксимацию (и интерполяцию) функции $y_i(x_i)$, заданной n -отсчетами (начиная с 1) по методу наименьших квадратов. Программа аналогична программе 5.25 и может быть проверена по контрольному примеру для программы 5.24.

§ П5.13. Приближение функций по Чебышеву

Среди многочленов, с помощью которых можно приближенно задать функцию $f(x)$, особое место занимают многочлены Чебышева $T_i(x)$ (см. § 3.4). Многочлены $T_i(x)$ формируются с помощью рекуррентного соотношения $T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x)$, при $T_0(x) = 1, T_1(x) = x$. Таким образом, получаем: $T_2(x) = 2x^2 - 1; T_3(x) = 4x^3 - 3x; T_4(x) = 8x^4 - 8x^2 + 1$ и т. д. Значения x должны быть в пределах $[-1, 1]$.

В технике приближения функций используется линейная комбинация многочленов Чебышева, которая записывается в виде составного многочлена $T(x) = a_0T_0(x) + a_1T_1(x) + \dots + a_nT_n(x)$, где a_0, a_1, \dots, a_n — коэффициенты Чебышева.

Приближение функции $f(x)$ с помощью многочлена $T(x)$ обеспечивает минимальную погрешность отклонения $T(x)$ от $f(x)$ (в сравнении с другими интерполяционными многочленами). Это достигается за счет оптимального выбора узлов интерполяции в заданном промежутке $[-1, 1]$ изменения аргумента x . Многочлен $T(x)$ может быть представлен в форме обычного степенного многочлена (полинома) $P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$.

Алгоритм преобразования линейной комбинации многочленов Чебышева в обычный степенной многочлен.

1. Вводим коэффициенты a_0, a_1, \dots, a_n многочлена $T(x)$ и образуем массив $a(i)$.

2. Делим a_0 на 2: $a_0 = a_0/2$.

3. Для $j=2, 3, \dots, n$ и $k=j, j+1, \dots, n$ проводим преобразования коэффициентов по формулам

$$a_{k-2} = a_{k-2} - a_k; \quad a_k = 2a_k.$$

В результате получаем коэффициенты a_i степенного полинома $P(x)$.

Алгоритм преобразования коэффициентов степенного полинома в коэффициенты линейной комбинации многочленов Чебышева.

1. Вводим коэффициенты степенного полинома a_i .

2. Для $j=n, n-1, \dots, 2$ и $k=j, j+1, \dots, n$ выполняем преобразования коэффициентов по формулам $a_k = a_k/2$; $a_{k-2} = a_{k-2} + a_k$.

3. Умножаем a_0 на 2: $a_0 = 2a_0$.

В результате получаем коэффициенты Чебышева для $T(x)$.

Программа П5.13/1.

```

10 PRINT 'ПРЕОБРАЗОВАНИЕ КОЭФФИЦИЕНТОВ МНОГОЧЛЕНОВ T(X) И P(X) .'
20 INPUT 'ВВЕДИТЕ СТЕПЕНЬ МНОГОЧЛЕНА N=':N:DIM A(N)
25 FOR I=0 TO N:PRINT!2.0!'ВВЕДИТЕ A' I:INPUT A(I):NEXT I
30 PRINT 'ВВЕДИТЕ КОД ИСХОДНОГО МНОГОЧЛЕНА'
40 INPUT '0-ДЛЯ T(X) И 1-ДЛЯ P(X) ' S:IF S=1 THEN 80
50 LET A(0)=A(0)/2:FOR J=2 TO N:FOR K=N TO J STEP-1
60 LET A(K-2)=A(K-2)-A(K):LET A(K)=2*A(K)
70 NEXT K:NEXT J:GOTO 130
80 FOR J=N TO 2 STEP-1:FOR K=J TO N
90 LET A(K)=A(K)/2:LET A(K-2)=A(K-2)+A(K)
100 NEXT K:NEXT J:LET A(0)=A(0)*2
110 IF K=1 THEN PRINT 'КОЭФФИЦИЕНТЫ МНОГОЧЛЕНА P(X) ':GOTO 130
120 PRINT 'КОЭФФИЦИЕНТЫ МНОГОЧЛЕНА T(X) '
130 FOR I=0 TO N:PRINT!2.0!'A' I=' ' !F1.9!A(I)
140 NEXT I:END

```

Работу программы можно проверить с помощью взаимного преобразования многочленов ($n = N = 4$):

$$T(x) = 8.75T_0(x) + 5T_1(x) + 4T_2(x) + 1T_3(x) + 0.625T_4(x),$$

$$P(x) = 1 + 2x + 3x^2 + 4x^3 + 5x^4.$$

Алгоритм приближения функции по Чебышеву.

1. Задаем степень n многочлена $T(x)$ и пределы a, b области изменения аргумента x функции $f(x)$.

2. Для $i=0, 1, \dots, n$ на отрезке $[-1, 1]$ формируем сетку оптимальных значений аргумента в узлах чебышевской интерполяции $x_i = \cos\left(\frac{2i+1}{n+1} \cdot \frac{\pi}{2}\right)$, приводим x_i в отрезок

$[a, b]$ $x_i = \frac{b+a}{2} + \frac{b-a}{2} x_i'$ и вычисляем значения $f(x_i)$.

3. Для $k=0, 1, \dots, n$ и $i=0, 1, \dots, n$ вычисляем

$$a_k = \frac{2}{n+1} \sum_{i=0}^n f(x_i) \cos\left[\frac{(2i+1)k\pi}{2(n+1)}\right].$$

В результате получаем коэффициенты a_0, a_1, \dots, a_n многочлена $T(x')$, приближающего функцию $f(x)$, т. е. аппроксимируем $f(x)$ многочленом $T(x')$.

Вычисление значений $T(x)$ выполняется по следующему алгоритму.

1. Считая заданным массив a_k , задаем память под массив из $(n+2)$ вспомогательных коэффициентов b_k . Полагаем $b_{n+2} = 0$ и $b_{n+1} = 0$.

2. Задаем значение x на отрезке $[a, b]$ и переводим его в отрезок $[-1, 1]$ с помощью преобразования $x' = \frac{(x - (b+a)/2) \cdot 2}{(b-a)}$.

3. Для $k=n, n-1, \dots, 1$ вычисляем $b_k = a_k - b_{k+2} + 2b_{k+1}x$.

4. Находим $T(x') = a_0/2 - b_2 + b_1x$.

Приведенная ниже программа обеспечивает приближение $f(x)$ многочленом $T(x')$ заданной степени n , вычисление a_0, a_1, \dots, a_n многочлена $T(x')$ или $P(x)$ (задается кодом 0 или 1) и вычисление значения $T(x')$ или $P(x)$.

Программа П5.13/2.

```

10 PRINT 'ПРИБЛИЖЕНИЕ ФУНКЦИИ ПО ЧЕБЫШЕВУ .'
20 INPUT 'ВВЕДИТЕ СТЕПЕНЬ МНОГОЧЛЕНА N=':N:DIM Y(N),A(N),B(N+2)
30 INPUT 'ВВЕДИТЕ ПРЕДЕЛЫ ИЗМЕНЕНИЯ X A,B 'A1,B1
40 LET Q=N+1:LET M=#PI/2/Q:LET C=(B1+A1)/2:LET D=(B1-A1)/2
50 FOR I=0 TO N:LET X1=COS((2*I+1)*M):LET X=C+D*X1
60 GOSUB 300:LET Y(I)=F:NEXT I
70 FOR K=0 TO N:FOR I=0 TO N
80 LET A(K)=A(K)+Y(I)*COS(K*(2*I+1)*M):NEXT I
90 LET A(K)=A(K)*2/Q:NEXT K
100 PRINT 'ЗАДАЙТЕ ВВОДОМ КОДА ВИД МНОГОЧЛЕНА'
110 INPUT '0-T(X), 1-P(X) 'S:IF S=1 THEN 150
120 PRINT 'КОЭФФИЦИЕНТЫ МНОГОЧЛЕНА T(X) '
130 FOR I=0 TO N:PRINT!2.0!'A' I=' ' !F1.9!A(I)
140 NEXT I:GOTO 190

```



```

150 LET A(0)=A(0)/2:FOR J=2 TO N:FOR K=N TO J STEP-1
160 LET A(K-2)=A(K-2)-A(K):LET A(K)=2*A(K):NEXT K:NEXT J
170 PRINT 'КОЭФФИЦИЕНТЫ МНОГОЧЛЕНА P(X)'
180 FOR I=0 TO N:PRINT!2.0!'A'I'='!F1.9!A(I):NEXT I
190 INPUT 'ВВЕДИТЕ X='X1:LETP=0:LET X=(X1-C)/D
200 IF S=0 THEN 240
210 FOR I=N TO 1 STEP-1:LETP=(P+A(I))*X
220 NEXT I:LETP=P+A(0)
230 PRINT 'P(X)='P:GOTO 190
240 LET B(N+2)=0:LET B(N+1)=0:FOR K=N TO 1 STEP-1
250 LET B(K)=A(K)-B(K+2)+2*B(K+1)*X:NEXT K
260 LET T=A(0)/2-B(2)+B(1)*X
270 PRINT 'T(X)='T:GOTO 190
300 REM 'ПОДПРОГРАММА ВЫЧИСЛЕНИЯ F=F(X)'
310 LET F=LOG(1.5+X/2):RETURN:END

```

Вычисление $f(x) = F$ оформляется подпрограммой, записанной со строки 300. Контрольный пример дан для функции $F = \ln(1.5 + x/2)$. Задав $n = N = 3$, $a = A = 0$ и $b = B = 3$, получим (код 0) следующие значения коэффициентов Чебышева: $a_0 = 1,563836218$, $a_1 = 0,3431444714$, $a_2 = -0,02942874414$ и $a_3 = 0,003307617869$. Для $x = 1$ $f(1) = 0,6932432419$, а для $x = 2$ $f(2) = 0,916371022$ (при точных значениях $f(1) = 0,6931471806$ и $f(2) = 0,9162907319$). Повторив вычисления при коде 1, получим коэффициенты степенного многочлена $P(x)$: $a_0 = 0,8113468529$, $a_1 = 0,3332216178$, $a_2 = -0,05885748827$ и $a_3 = 0,01323047147$. Значения $f(1) = P(1)$ и $f(2) = P(2)$ при этом совпадают с указанными выше. С увеличением n предшествующие значения a_i многочлена $T(x)$ сохраняются (точнее,

из-за погрешности вычислений они слабо меняются).

Из-за неравномерного и труднопредсказуемого расположения узлов на $[a, b]$ метод Чебышева обычно не применяется для интерполяции табличных данных. Он очень удобен при задании $f(x)$ в аналитической форме, но может быть применен и для аппроксимации графически заданных функций. В последнем случае подпрограмму записывают в виде

```

310 PRINT 'ДЛЯ X='X
320 INPUT 'ВВЕДИТЕ ЗНАЧЕНИЕ F(X) 'F:
RETURN:END

```

По подсказанным программой значениям x вводятся снятые с графика значения $f(x)$

1. Анго А. Математика для электро- и радиоинженеров.— М.: Наука, 1965.
2. Бал И. Дж., Гарт Р. Простые и точные формулы для несимметричной полосковой линии с конечной толщиной полоски // ТИИЭР, 1977, 65, № 11, с. 104.
3. Батищев Д. И. Методы оптимального проектирования.— М.: Радио и связь, 1984.
4. Бахвалов Н. С. Численные методы.— М.: Наука, 1975.
5. Бронштейн И. Н., Семендяев К. А. Справочник по математике для инженеров и учащихся втузов.— М.: Наука, 1986.
6. Волков Е. А. Численные методы.— М.: Наука, 1982.
7. Горинштейн А. М. Практика решения инженерных задач на ЭВМ.— М.: Радио и связь, 1984.
8. Дафдет-Смит П. Практическая астрономия с калькулятором/Пер. с англ.; Под ред. Г. А. Лейкина.— М.: Мир, 1982.
9. Дьяконов В. П. Расчет нелинейных и импульсных устройств на программируемых микрокалькуляторах.— М.: Радио и связь, 1984.
10. Дьяконов В. П. Справочник по расчетам на микрокалькуляторах.— М.: Наука, 1985.
11. Дьяконов В. П. Персональные ЭВМ в аппаратуре и технике эксперимента//Приборы и техника эксперимента, № 1, 1986, с. 7.
12. Иванов В. И., Иванов Е. А., Муренко Л. П. Малогабаритные вычислительные комплексы индивидуального пользования//Радиотехника, 1983, № 1, с. 24.
13. Иоссель Ю. Я., Кочанов Э. С., Струнский М. Г. Расчет электрической емкости.— Л.: Энергоиздат, 1981.
14. Иоффе А. Ф. Массовые персональные ЭВМ серии «Агат»//Микропроцессорные средства и системы, 1984, № 1, с. 56.
15. Калантаров П. Л., Цейтлин Л. А. Расчет индуктивностей.— Л.: Энергия, 1970.
16. Кетков Ю. Л. Программирование на бейсике.— М.: Статистика, 1978.
17. Клингман Э. Проектирование микропроцессорных систем/Пер. с англ.; Под ред. С. Д. Пашкеева.— М.: Мир, 1980.
18. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров.— М.: Наука, 1984.
19. Косенков С. М., Полоскин А. Н., Счепницкий З. А., Дябин М. И., Половянюк А. И. Бытовая персональная микроЭВМ «Электроника БК-0010»//Микропроцессорные средства и системы, № 1, 1985, с. 22.
20. Кузнецов Е. Ю., Острецов Б. В., Минкин Л. К., Егорова Ю. И. Микрокалькуляторы.— М.: Радио и связь, 1984.
21. Кухлинг Х. Справочник по физике/Пер. с нем.; Под ред. Е. М. Лейкина.— М.: Мир, 1982.
22. Мейнке Х., Гундлах Ф. Радиотехнический справочник. Т. 1.— М.: Госэнергоиздат, 1961.
23. Мошиц Г., Хорн П. Проектирование активных фильтров/Пер. с англ.; Под ред. И. Н. Теплока.— М.: Мир, 1984.
24. Набор программ для ЭЦВМ «Мир». Т. 1—2.— Киев: Наукова думка, 1973.
25. Персональные ЭВМ//ТИИЭР, 1984, 72, № 3.
26. Погорелый С. Д., Слободянюк Т. Ф. Программное обеспечение микропроцессорных систем: Справочник.— Киев: Техника, 1985.
27. Попов А. А., Хохлов М. М., Глухман В. Л. Диалоговые вычислительные комплексы «Электроника НЦ-80-20»//Микропроцессорные средства и системы, 1984, № 4, с. 61.
28. Программирование на языке бейсик-плюс для СМ-4/В. П. Семьяк, Б. Р. Моздзювич, Д. П. Непочатых и др.— М.: Финансы и статистика, 1982.
29. Проектирование спускаемых автоматических космических аппаратов: Опыт разработки диалоговых процедур/Под ред. В. М. Ковтуненко.— М.: Машиностроение, 1985.
30. Прудников А. П., Брычков Ю. А., Маричев О. И. Интегралы и ряды.— М.: Наука, 1981.
31. Расчет электрических цепей и электромагнитных полей на ЭВМ/Под ред. Л. В. Данилова и Е. С. Филиппова.— М.: Радио и связь, 1983.
32. Самарский А. А. Введение в численные методы.— М.: Наука, 1982.
33. Сергеев Т. О. НР-85 — настольная ЭВМ индивидуального пользования//Зарубежная радиоэлектроника, 1981, № 10, с. 86.

34. Сигорский В. П., Петренко А. И. Основы теории электронных схем.— Киев: Внща' школа, 1971.
35. Справочник по расчету и проектированию АРС-схем/Букашкин С. А., Власов В. П., Змий Б. Ф. и др.; Под ред. А. А. Лайна.— М.: Радио и связь, 1984.
36. Справочник по специальным функциям/Пер. с англ.: Под ред. М. Абрамовица и И. Стиган.— М.: Наука, 1979.
37. Трохименко Я. К., Любич Ф. Д. Радиотехнические расчеты на микрокалькуляторах.— М.: Радио и связь, 1983.
38. Трохименко Я. К., Любич Ф. Д. Инженерные расчеты на программируемых микрокалькуляторах.— Киев: Техника, 1985.
39. Уорт Т. Программирование на языке бейсик.— М.: Машинное строительство, 1981.
40. Усилители с полевыми транзисторами/В. М. Немчинов, В. Г. Никитаев, М. А. Ожогин, В. В. Ляхович; Под ред. И. П. Степаненко.— М.: Сов. радио, 1980.
41. Щуп Т. Решение инженерных задач на ЭВМ.— М.: Мир, 1982.
42. Янке Е., Эмде Ф., Леш Ф. Специальные функции: Формулы, графики, таблицы.— М.: Наука, 1977.
43. CASIO FX-702P. Instruction Manual. BM Mark for high quality & Reliability. Casio.

- Алгебраические операции 41
 — уравнения 93
 Алгоритм 43
 Алфавит бейсика 21
 Амплитудно-частотная характеристика (АЧХ) 132
 Ангера функция 155
 Аппроксимация полиномиальная 82, 234
 — методом наименьших квадратов 142
 — — — с заданной погрешностью 143
 Асимметрии коэффициент 134
 Атмосферное давление 216
 Атенюаторов расчет 164
- Бейсик 21
 Бесселя функции 151
 — — модифицированные 151
 — — сферические 152
 Быстрое преобразование Фурье 125
- Вебера функция 156
 Векторный анализ 71
 Вероятности интеграл 157
 — функций распределения 158
 Версии бейсика 27
 Вклада вычисление 217
 Вращения метод 75
 Время счета 46
 Волновое сопротивление линий 174
 Вычислительные микросистемы 16
- Гамма-функция 149
 — неполная 150
 Гармоник коэффициент 128
 Гармонический синтез 116
 Гармоническое среднее 135
 Гаусса метод 75, 106.
 Геометрическое среднее 135
 Гиперболическая регрессия 140, 228
 Гиперболические функции 27, 59
 Гипергеометрическая функция 156
 Гистограмма 135
 Горнера схема 61
 Графиков вывод 49
- Данилевского метод 118
 Двойных интегралов вычисление 108
 Деление комплексных чисел 56
 — многочленов 62
 — отрезка в заданном отношении 73
 — с произвольной точностью 55
 Деления пополам метод 88, 97
 Делитель согласующий 163
 День недели 219
 Диалог 48
 Дилогарифм 156
 Диск магнитный 20
 Дисковый накопитель 10, 20
 Дисперсия 134
 Дисплей 10
 Дифференциальные уравнения 111
 Дифференциальных уравнений системы 111
 — — —, решение методом Рунге — Кутта 112
 — — —, — Рунге — Кутта — Мерсона 114
 — — —, — Рунге — Кутта — Фельберга 115
 — — —, — трапеций 111
- Дифференцирование численное 100
 Дихотомии метод 88, 97
 Дробно-рациональная функция 62
 Дробь цепная 51
 Дроссель фильтра 170
- Емкость между шарами 173
 — пленочных конденсаторов 172
 — проводников 172
- Заломинающее устройство 10
 — — оперативное (ОЗУ) 10
 — — постоянное (ПЗУ) 10
 Звезда — треугольник 162
 Зейделя метод 91, 226
 Зиверта функция 157
 Золотого сечения метод 97
- Индуктивность катушек 166
 Интеграл вероятности 157
 Интегральные показательные функции 147
 Интегральный косинус 149
 Интегральный синус 148
 Интегрирование численное 104
 — — методом Болдэ 105
 — — — Гаусса 106
 — — — Ньютона — Котеса 106
 — — — Симпсона 105
 — — — Чебышева 106
 — — — Уэддла 106
 — — разрывных функций 109
 Интерполяция 79
 — функций двух переменных 84
 Интерпретатор бейсика 21
- Касательных метод 86
 Каскад усилительный 180
 — — на биполярных транзисторах 181
 — — на полевых транзисторах 182
 Квадратичная интерполяция 85
 — — при минимизации функций 96, 99
 — — при решении нелинейных уравнений 90
 Квадратное уравнение 93
 Кельвина функции 157
 Ключ на мощном МДП-транзисторе 209
 Комбинаторика 70
 Комплексных чисел функции 56
 Контур резонансный 165
 Координат преобразование 42, 71
 Координатного спуска метод 99
 Кори алгебраических уравнений 93
 — нелинейных уравнений 86
 — полинома 93
 — — с комплексными коэффициентами 73
 — — системы нелинейных уравнений 91
 Котельникова ряд 117
 Коэффициент ранговой корреляции 138
 — чувствительности 101
 Коши задача 111
 Кубическое уравнение 98
- Лагранжа формулы интерполяции 80
 Линейная регрессия 138
 Лнейных уравнений системы 75
 Линии искусственные 179

Линии коаксиальные 174
— полосковые 177
— проводные 176
— спиральные 176
Логарифмическая производная гамма-функции 150
— регрессия 142

Массивы данных 35
Матрицы 65
Матричные операции 40, 65, 232
Моделирование ключа на мощном МДП-транзисторе 209
— релаксационного генератора 212
— статистическое 136
— трансформаторного усилителя мощности 207
Моменты начальные 133
Мюнте-Карло метод 136

Наибольший общий делитель 55
Наименьших квадратов метод 138, 228
Нелинейная регрессия 139
Нелинейных уравнений корни 86
Ньютона метод модифицированный 87
— — обобщенный 93

Обратной интерполяции — экстраполяции метод 90
Обращение матриц 68, 233
Операторы бейсика 21
Определителей вычисление 66
Ортогональные многочлены 63
Отладка программ 52
Отрезка длина 72

Параметры плоскости 73
— прямой 72
Парной корреляции коэффициент 137
Перевод программ 51
Переменные бейсика 21
Переходы безусловные 25
— с косвенной адресацией 27, 30
— условные 25
Периферийное оборудование 19
Персональные ЭВМ 9
Подпрограмма 45
Полинома значения 61
— корни 94
Поразрядного приближения метод 89, 91
Преобразование чисел по основанию 54
— обыкновенной дроби в цепную 54
— цепной дроби в обыкновенную 54
Приоритет операций 22, 35
Программирование на бейсике 43, 48
Программы линейные 43
— разветвляющиеся 44
— циклические 44
Производных вычисление 100
Простых итераций метод 87, 225, 228

Разложение функций в ряд 45, 147
— числа на простые множители 55
Регрессионный анализ 138
Редактирование программ 52
Рыбакова метод 88

Сглаживание данных эксперимента 144
Секущих метод 89
Секущих — хорд метод 89
Случайные числа 136
Собственные значения и векторы матриц 117
Сортировка данных 218
Сочетания 70
Спектральный анализ методом Берга 133
— — — двенадцати ординат 130
— — — степенного полинома 129
— — — пяти ординат 128
— — — семи ординат 128
— — параллельный 124
— — последовательный 123
— — при аналитически заданной функции 229
— — разрывных функций 131
— — финитных функций 123
Специальные функции 147
Сплайн-интерполяция 83, 231
Среднее арифметическое и геометрическое 135
Стандартное отклонение 134
Статистические расчеты 133
Струве функции 155
Суммирование данных таблицы 219

Температуры распределение 216
Трансцендентных уравнений решение 86
Трапеций метод 104, 114

Угол между векторами 73
— — вектором и осями координат 72
Умножение многочленов 63
Устойчивость многочлена 61

Фазо-частотная характеристика (ФЧХ) 62, 122
Факториал 70
Физические параметры транзистора 180
Фильтры активные 186
— — с большой добротностью 201
— — с малой добротностью 189
— — со средней добротностью 194

Френеля интегралы 153
Функции комплексного переменного 56
Функция $N!!$ 70

Хорд метод 89

Циклы 44

Частные производные 102
Числа Фибоначчи 56

Шина адресов и данных 9

Эйлера — Коши метод 111
— — — с итерациями 111
— метод модифицированный 111
Эйри функции 153
Эйткена метод интерполяции 80
Эйткена — Стеффенсона метод 90
Экстремумов поиск 96
Экстремума коэффициент 134
Эллиптические интегралы 154

Якоби метод 121

Страница	Строка, программа или формула	Напечатано	Должно быть, или, что следует сделать читателю
14	9 строка сверху	32 (дважды)	16 (дважды)
45	Программа	Лишние строки 60 и 70	Удалены строки 60 и 70
64	3 строка снизу	$x=3$	$x=2$
64	14 строка снизу	$= -0,26$	$= -0,296$
67	Программа 3.53	В строке 80 NEXT J: NEXT I	Перенести в начало строки 90
69	Программа 3.56	В строке 160 NEXT M	Перенести на строку 165
70	5 строка сверху	81	60
70	Программа 3.60	Лишняя строка 15	Удалить строку 15
71	13 строка сверху	$Y=R \sin \varphi \cos Q$	$Y=R \sin \varphi \sin Q$
85	2 строка сверху	7 (b)	6 (b)
94	Программа 4.23	В строке 330 GOTO 110	GOTO 120
99	Программа 4.29	Лишняя строка 135	Удалить строку 135
110	Формула (4.32)	$f(x, y)$	$f(x, y) dx dy$
120	Программа 4.59	В строке 190 I	Заменить на B
122	Формулы (5.5), (5.6), (5.9), (5.10)	Формулы с пропущенными знаками dt	Знаки dt в конце формул (5.5), (5.6), (5.9) и (5.10)
126	Программа 5.3	В строке 50 X (I), Y (I)	X (I+10), Y (I+10)
132	Программа 5.9	В строке 280 P=W*K	W=P*K
146	5 строка снизу	0,70114285714	0,7014228574
146	4 строка снизу	1,08714286	1,080714286
152	Программа 6.14	В строке 170 GOTO 20	GOTO 30
159	2 строка сверху	$=0,8413447497$	$=0,84270079$
159	Программа 6.27	В строке 20 LETG=1	Добавить в начало строки LETD=1
159	Программа 6.27	В строке 50 LETG=G*D/X В строке 55 EXP(A)	Перенести в начало строки 50 EXP(B)
159	28 строка сверху	$Q(1,1)=0,05121793333$	0,3173105079
164	14 строка сверху	$Z=R+j\omega L$	$Z=R+j(\omega L - 1/\omega C)$
182	31 строка снизу	$R_n=34,946 \text{ Ом}$ $a=2,534 \cdot 10^5 \text{ Ом}$	$R_n=5000 \text{ Ом}$, $a=0,98$, $R_{вх}=34,946 \text{ Ом}$, $R_{вых}=253428,6 \text{ Ом}$
209	1 строка снизу	$C_{зн}$	$C_{зн}$
229	Программа П5.9	В строке 120 (Z*Z/N0)	Z*Z*N0

59 **Програма 3.23** (читать так)

```

0 PRINT 'ВЫЧИСЛЕНИЕ F(Z)=ATN(Z)'
1 INPUT 'ВВЕДИТЕ RE Z='A: INPUT 'ВВЕДИТЕ IM Z='B
2 LET E=ATN((Z*A/(1-A*A-E*B))/2
3 LET F=LOG((A*A+(B+1)^2)/(A*A+(B-1)^2))/4
4 PRINT 'RE F(Z)='E: PRINT 'IM F(Z)='F
5 GOTO 20: END

```

1 p. 40 к.

