



На современном этапе технической революции все большее значение приобретают вопросы научной организации труда и управления, экономики промышленного производства, автоматизации технологических процессов с применением электронных вычислительных машин, психологии труда и др.

Сегодня каждый инженер, независимо от его узкой специализации, должен не только владеть основами знаний в этих отраслях, но и быть информированным о последних достижениях в них.

Именно эти вопросы определяют тематику «Библиотеки инженера»

Kned . . . & Texnika

180

Я. К. ТРОХИМЕНКО, Ф. Д. ЛЮБИЧ

ИНЖЕНЕРНЫЕ

РАСЧЕТЫ

НА МИКРОКАЛЬКУЛЯТОРАХ

ББК 32. 973
6Ф7.3
Т76

T76 Трохименко Я. К., Любич Ф. Д.
Инженерные расчеты на микрокалькуляторах.—К.:
Техника, 1980. 383 с., ил. — (Б-ка инженера). — Би-
лиогр.: с. 380—381

В пер.: 1 р. 70 к. 20 000 экз.

В книге изложено математическое и программное обеспечение ин-
женерно-технических расчетов на электронных микрокалькуляторах.
Рассмотрены типовые задачи обработки экспериментальных дан-
ных, вычисления функциональных зависимостей и специальных функ-
ций, решения нелинейных и дифференциальных уравнений, числен-
ное интегрирование, практический гармонический анализ и цифро-
вое моделирование. Приведено свыше 200 программ решений типо-
вых задач на программируемых микрокалькуляторах.

Рассчитана на широкий круг инженерно-технических работников.

Т 30502—091
M202(04)-80 11.80.2405000000

ББК 32.973
6Ф7.3

Рецензенты д-р техн. наук Л. Я. Нагорный, Ю. М. Польский

Редакция литературы по энергетике, электронике, кибернетике и связи
Зав. редакцией З. В. Божко



Издательство «Техника», 1980

ПРЕДИСЛОВИЕ

Решения ХХV съезда КПСС предусматривают бурное развитие вычислительной техники и ее широкое внедрение во все области народного хозяйства. Автоматизируя математические операции с помощью электронных вычислительных машин (ЭВМ), можно существенно сократить сроки, повысить точность и, в конечном итоге, снизить стоимость инженерного проектирования, экономических расчетов и научно-технических исследований.

Экономическая целесообразность использования ЭВМ различной производительности определяется размерами решаемых задач. Эксплуатация ЭВМ высокой производительности требует больших материальных затрат, непосредственный доступ инженера к таким машинам ограничен, а подготовка исходных данных и отладка программ связаны со значительными затратами времени. Между тем большинство инженерных расчетов относительно просты или их можно разбить на простые части, решение которых с большей экономической эффективностью обеспечивают ЭВМ меньшей производительности. Такие сравнительно несложные задачи наиболее оперативно решаются с помощью малогабаритных электронных клавишных вычислительных машин, отличающихся удобством ввода и вывода информации и не требующих специальной подготовки их пользователей. К этим машинам относятся и портативные микроЭВМ — электронные микрокалькуляторы.

С помощью арифметического микрокалькулятора, выполняющего операции сложения и умножения чисел, в принципе, можно решать те же задачи, что и с помощью самых совершенных универсальных ЭВМ. Однако чем сложнее задача, тем больше времени требуется для ее решения с помощью такого микрокалькулятора. В инженерной практике более эффективны электронные микрокалькуляторы, выполняющие не только арифметические операции, но и вычисляющие автоматически элементарные или часто встречающиеся функции по заданным значениям аргументов. Наиболь-

шие возможности автоматизации инженерных расчетов связаны с программируемыми электронными микрокалькуляторами, практически воплощающими еще недавно казавшуюся фантастической мечту об ЭВМ в кармане. Использование программируемого режима работы таких микрокалькуляторов при решении задач, связанных с многократными однотипными вычислениями, существенно сокращает затраты времени на расчет.

Значение «малой» автоматизации инженерно-технических расчетов с помощью микрокалькуляторов, учитывая многочисленность их пользователей, трудно переоценить. Вместе с тем, эффективность их применения, как и ЭВМ любого другого класса, существенно зависит от умения пользователя составить алгоритм или программу вычислений таким образом, чтобы получить искомый результат с требуемой или возможной точностью при минимальных затратах времени.

В книге рассматриваются особенности использования микрокалькуляторов любых типов — от простейших арифметических, таких как «Электроника Б3-04», «Электроника Б3-24» и других, до программируемых типа «Электроника Б3-21» и «Электроника Б3-34» — для решения широкого круга задач. Приведено свыше 200 программ автоматических вычислений, сопровождаемых контрольными примерами, составленных на входном языке программируемого микрокалькулятора типа «Электроника Б3-21», которые достаточно просто могут быть переведены на входные языки программируемых микрокалькуляторов других типов как выпускаемых в настоящее время, так и тех, которые, несомненно, появятся в ближайшем будущем. В связи с этим хотелось бы, чтобы читатель рассматривал настоящую работу не как инструкцию по эксплуатации микрокалькуляторов упоминаемых типов, а как справочное пособие по алгоритмам решения основных инженерных задач и их реализации на микрокалькуляторах. Авторы будут считать, что они достигли своей цели, если знакомство с возможностями микрокалькуляторов вызовет у читателя интерес к вопросам численного решения задач и программированию.

В приведенных в книге примерах условия инженерных задач сформулированы в основном в терминах электротехники, радиотехники и электроники, но методика и программы решения таких задач при аналогичной математической постановке в равной мере пригодны для любой другой области науки и техники.

Отзывы и пожелания о книге просим направлять по адресу: 252601, Киев, 1, ГСП, Крецатик, 5, издательство «Техника».

Глава 1

ЭЛЕМЕНТЫ ИНЖЕНЕРНЫХ ВЫЧИСЛЕНИЙ

1. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ИНЖЕНЕРНЫХ ЗАДАЧАХ

Основная задача инженера заключается в разработке способов построения физических объектов с заданными свойствами из существующих компонентов с известными свойствами. Результат решения подобной задачи, называемый проектом, в общем случае содержит описание способа соединения компонентов выбранного типа (структурной схемы), а также описания предполагаемой конструкции и способа (технологии) изготовления проектируемого объекта. Так как свойства физических объектов описываются числами, то процесс инженерного проектирования связан с операциями над математическими моделями, отображающими посредством условных символов числа и отношения между ними.

Свойство или проявление физического* явления математически описывается (моделируется) некоторым числом y , полученным в результате наблюдений или измерений. Зависимость этого числа от условий, в которых происходит рассматриваемое явление, описывается функцией

$$y = y(t), \quad (1)$$

где t — независимая координата отсчета.

Физические процессы протекают во времени, поэтому обычно в качестве координаты отсчета выбирают отрезки времени от некоторого начального момента t_0 , однако эта координата может иметь и другую размерность.

Во многих случаях удобно моделировать свойства явлений функциями комплексного аргумента. Наиболее часто используют отображение функций вещественного аргумента $y(t)$ в частотную область аргумента $j\omega$ с помощью преобразования Фурье

$$f(j\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t} dt = |f(j\omega)|e^{-j\phi(\omega)}. \quad (2)$$

* Здесь и далее слово «физический» используется как синоним слов «реально существующий».

Комплексная функция $f(j\omega)$, называемая спектральной плотностью, для периодических функций $f(t)$ вырождается в дискретный спектр в виде бесконечного в общем случае множества гармонических составляющих.

В физическом мире любое явление есть причина и следствие других явлений. Однако причиной явления со свойством, описанным переменной $y(t)$, является не координата отсчета t , а некоторое свойство того же или другого явления, описанное переменной $x(t)$, изменяющейся в той же системе отсчета. Причинно-следственная связь между двумя явлениями или их свойствами, описанными переменными $x(t)$ и $y(t)$ и соответственно рассматриваемыми как причина (воздействие) и следствие (реакция), математически описывается некоторой зависимостью

$$y(t) = y(x(t)) \equiv ax(t). \quad (3)$$

Сложную функцию $y(x(t))$ обычно называют динамической (или временной, если t имеет размерность времени) характеристикой канала связи между воздействием и реакцией. Свойства этого канала характеризуются также передачей или, если она равна постоянному числу, параметром $a = y(x(t))/x(t)$. Формально в канале причинно-следственной связи над переменной $x(t)$ воздействия выполняется некоторая операция, результатом которой является переменная $y(t)$. Поэтому букву a в уравнении (3) можно рассматривать как символ оператора, ставящего число $y(t)$ в соответствие числу $x(t)$. Математическая модель этого оператора, определяющая способ вычисления $y(t)$ по заданному значению $x(t)$, зависит от свойств конкретного канала причинно-следственной связи.

В связи с материальностью каналов причинно-следственных связей им присущи все основные свойства каналов передачи энергии — нелинейность, активность, инерционность, распределенность и случайность параметров.

Энергия, поступающая от источника воздействия $x(t)$, взаимодействует с каналом связи и изменяет его свойства, что проявляется в зависимости передачи $a = a(x(t))$ от уровня воздействия и в нелинейности уравнения (3) причинно-следственной связи.

Свойства канала связи зависят также от сторонних воздействий $z(t)$, не учитываемых явно в формуле (3). Поэтому в общем случае передача канала связи является сложной функцией $a = a(x(t), z(t))$, а уравнение (3) — параметрическим. При этом реакция поддерживается энергией, поступающей не только от источника воздействия $x(t)$, но и от внутреннего для данного канала источника $z(t)$. Поэтому каналы связи, при моделировании которых нельзя преобречь влиянием сторонних воздействий, называются активными или параметрическими.

Всеобщая взаимосвязь явлений проявляется в зависимости $y(t)$ от бесконечного числа сторонних воздействий, что приводит к случайным изменениям передачи канала рассматриваемой причинно-следственной связи.

Процессы накопления энергии в канале связи приводят к его инерционности, проявляющейся в зависимости реакции от значений воздействия не только в данный, но и в предыдущие моменты времени. Поэтому в общем случае реакция канала равна результату решения интегрально-дифференциального (относительно независимой переменной t) уравнения при начальных условиях, определяемых для начального значения t_0 :

$$y(t) = y(x(t), z(t), t) \equiv a(y(t), x(t), z(t), t) x(t). \quad (4)$$

Ограничность скорости распространения энергии проявляется в задержке во времени и распределении реакции (отображаемому распределением параметров) вдоль протяженных каналов причинно-следственной связи, что существенно усложняет их математическое описание. В большинстве практических случаев можно пренебречь влиянием некоторых из рассмотренных свойств каналов причинно-следственной связи. В частности, каналы небольшой протяженности можно рассматривать как сосредоточенные, что значительно упрощает их моделирование. При достаточно медленных изменениях воздействий инерционностью причинно-следственной связи пренебрегают и описывают ее статической характеристикой

$$y = y(x, z) \equiv a(x, z)x.$$

Следовательно, свойства безынерционных каналов связи зависят только от уровня воздействий.

По отношению к достаточно малым изменениям уровня воздействий (слабым воздействиям) свойства канала связи не зависят от уровня этих изменений, а уравнение (3) линейно. Для определения передачи линейных инерционных каналов связи можно использовать любые пробные воздействия удобного вида. В частности, для определения передачи таких каналов во временной области используют пробные воздействия в виде единичной функции ($x(t) = 1$ при $t \geq 0$ и $x(t) = 0$ при $t < 0$) или единичного импульса с бесконечно малой длительностью, интеграл которого равен единице. Реакции на эти воздействия называют соответственно переходной и импульсной характеристиками. Зная эти характеристики, можно по определенным правилам вычислить реакцию линейного канала на воздействие любого вида.

Операторное уравнение (3) для линейного инерционного канала представляет собой решение (4) линейного интегрально-дифференциального уравнения аргумента t . При отображении последнего в частотную область уравнение (3) преобразуется в алгебраическое уравнение

$$y(j\omega) = a(j\omega)x(j\omega), \quad (5a)$$

где

$$a(j\omega) = a(j\omega)e^{j\varphi(\omega)} = \operatorname{Re}a(j\omega) + j\operatorname{Im}a(j\omega) \quad (5b)$$

— оператор канала связи, называемый частотной характеристикой;
 $x(j\omega)$ — спектральное представление (2) воздействия.

Для экспериментального определения частотной характеристики $a(j\omega) = y(j\omega)/x(j\omega)$ изменяют частоту ω пробного воздействия $x = x_m \sin \omega t$, одновременно измеряя реакцию $y(j\omega)$ на это воздействие.

Операторное уравнение (3) для линейного инерционного канала отображают также в область комплексной частоты $p = \sigma + j\omega$ алгебраическим уравнением

$$y(p) = a(p)x(p), \quad (6a)$$

где $x(p)$ — изображение по Лапласу [14] оригинала $x(t)$. Оператор передачи в этом случае отображается дробно-рациональной функцией

$$a(p) = \frac{\sum_{k=0}^m b_k p^k}{\sum_{k=0}^n c_k p^k} = \frac{d \prod_{k=1}^m (p - p_{ok})}{\prod_{k=1}^n (p - p_{pk})} \quad (6b)$$

с вещественными коэффициентами c_k , b_k и d и вещественными или комплексно-сопряженными корнями числителя (нулями) и знаменателя (полюсами).

При подстановке $p = j\omega$ уравнение (6a) преобразуется в уравнение (5a), а функция (6b) — в частотную характеристику (5b). Оригинал функции (6b) равен импульсной характеристике моделируемого канала связи, а оригинал $a(p)/p$ — его переходной характеристике. Для нелинейных инерционных каналов подобные представления непосредственно неприменимы, что существенно усложняет моделирование, так как приходится составлять и решать нелинейные дифференциальные уравнения.

Результаты экспериментальных измерений зависимостей вида $y(x)$, моделирующих воздействия, реакции или операторы каналов связи после статистической обработки отображают удобными для последующего использования графическими, табличными, буквенными или алгоритмическими математическими моделями.

Графическая модель представляет собой непрерывный график зависимости $y(x)$ в заданном интервале x , тогда как табличная образуется упорядоченным множеством пар чисел x_i и $y_i = y(x_i)$ для ряда точек $i = 0, 1, \dots, n$ в интервале изменения x . Последняя отображает функцию лишь для конечного числа дискретных значений x_i , и для вычисления функции в остальной части заданного интервала аргумента используют различные способы интерполяции.

Буквенные модели составляют в виде математических формул, содержащих символы чисел и операций над ними, включая общепринятые условные обозначения функций и функционалов определенных типов. Буквенные модели являются частным случаем алгоритмических моделей в виде словесно-формульных описаний,

схем или машинных программ выбранного способа (алгоритма) вычисления значений y по заданным значениям x .

Причина x следствия $y = y(x) \equiv a_1x$ может быть следствием $x = x(q) \equiv a_2q$ некоторого явления с переменной q . Связь между y и q в этом случае описывается зависимостью $y = y(x(q)) \equiv aq$, где $a = a_1a_2$ — оператор эквивалентного канала связи (рис. 1, а). Подобные цепочки причинно-следственных связей могут образовать замкнутые контуры, подобные показанному на рис. 1, б, где $x_1 = x_1(q, x_2) \equiv \Theta q + \beta x_2$; $x_2 = x_2(x_1) \equiv \mu x_1$, а связь между x_2 и q отображается эквивалентным каналом связи с передачей κ , определяемой операторным уравнением

$$x_2(q) = kq \equiv \Theta\mu q/(1 - \beta\mu). \quad (7)$$

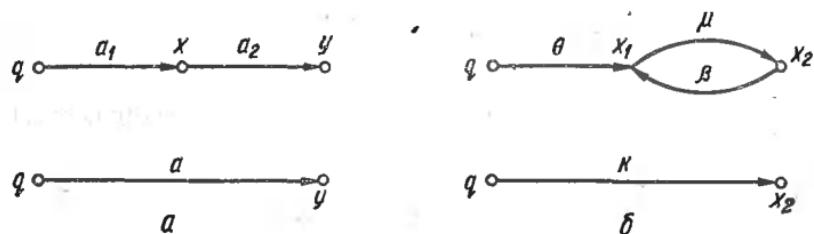


Рис. 1. Графы цепочек причинно-следственных связей:
а — разомкнутых; б — замкнутых

Если хотя бы один из каналов с передачей μ или β активен, то в моделируемом объекте возникает физическое явление, называемое обратной связью и заключающееся в зависимости причины x_1 следствия x_2 от этого следствия. При $\beta\mu > 1$ эта обратная связь называется положительной. В системе причинно-следственных связей с петлей (контуром) положительной обратной связи при $\beta\mu = 1$ возникает неустойчивость (самовозбуждение), при котором явления с переменными x_2 и x_1 поддерживаются энергией, поступающей от активного канала, а не от источника воздействия q .

В общем случае множество причинно-следственных связей между n явлениями с переменными x_i ($i = 1, 2, \dots, n$) можно рассматривать как автономную физическую систему* (объект или ситуацию), в которой эти явления поддерживаются энергией, поступающей от независимых воздействий q_i через входы (места определения переменных x_i) системы и от сторонних воздействий z_a через активные каналы связи. Математической моделью такой системы может служить система из n в общем случае операторных уравнений

$$\sum_{i=0}^n a_{ij}x_i = q_i; \quad i = 1, 2, \dots, n \quad (8a)$$

* Системой называют множество связанных элементов. Система характеризуется структурой, определяющей отношения между элементами, и параметрами (операторами), описывающими свойства элементов.

$$AX = Q, \quad (86)$$

где A — квадратная матрица операторов (передач) каналов, связывающих переменные x_i , X и Q — векторы-столбцы переменных x_i и q_i соответственно.

Модель проектируемого объекта в виде системы уравнений (8) может содержать подсистемы (блоки) уравнений, отображающих не только непосредственные энергетические связи между входами объекта, но и причинно-следственные связи между производственно-экономическими, надежностными и другими требуемыми показателями (параметрами).

Систему уравнений (8) можно преобразовать в другую эквивалентную (отображающую с такой же точностью свойства моделируемого объекта) математическую модель в виде системы уравнений, графа, схемы замещения или частично-упорядоченного множества [25].

В приведенном рассмотрении математических моделей в качестве исходных использовались лишь положения о причинности и материальности окружающего нас мира. Поэтому эти модели пригодны для описания любых физических систем — как объектов в виде приборов или устройств, так и физических (реально существующих) ситуаций, связывающих некоторые исходные состояния (причины) с последующими состояниями (следствиями) системы.

Свойства проектируемого и еще не существующего в природе объекта можно определить только по его математической модели, составленной из математических моделей существующих частей (компонентов) объекта с известными свойствами. Поэтому инженерное проектирование в основном сводится к операциям над математическими моделями физических систем.

При схемотехническом проектировании инженер по данным технического задания выбирает начальный вариант схемы, отображающей способ соединения компонентов (структуру) объекта, а также параметры этих компонентов из некоторого допустимого множества. Заменяя структурную схему более удобной для вычислений математической моделью (обычно в виде системы уравнений), инженер рассчитывает по ней характеристики, определяющие свойства проектируемого объекта с выбранной структурной схемой и предполагаемыми параметрами компонентов. При их несоответствии требованиям технического задания инженер изменяет параметры компонентов (при сохранении схемы их соединения) до совпадения рассчитываемых характеристик с требуемыми. Если этого не удается добиться при наилучшем (оптимальном) сочетании допустимых значений параметров компонентов, инженер изменяет схему объекта с одновременной оптимизацией параметров после каждого шага изменения структуры. Подобные операции в цикле оптимизации структуры объекта, охватывающем циклы оптимизации параметров (рис. 2), продолжаются до полного удовлетворения требо-

ваний технического задания. Завершается проектирование составлением документации, описывающей полученную структурную схему объекта.

Рассмотренная обобщенная схема с циклами оптимизации структуры и параметров характерна и для инженерного проектирования конструкции и технологии изготовления заданного объекта. Иногда структуру и параметры схемы определяют по справочным данным и расчетным формулам для типовых решений, но выбор возможных вариантов решения также соответствует оптимизационным циклам.

Вычисления на этапах анализа и оптимизации математической модели проектируемого объекта можно ускорить при использовании ЭВМ. Однако применение ЭВМ высокой производительности с большой стоимостью машинного времени экономически оправдано лишь при решении достаточно сложных задач. Большинство повседневных инженерных задач экономически целесообразно решать с помощью малых ЭВМ с низкой стоимостью машинного времени, в частности — портативных микро-ЭВМ, называемых микрокалькуляторами.

Независимо от используемых вычислительных средств решение инженерных задач включает три основных взаимосвязанных этапа выбора:

1) наиболее удобной для последующих вычислений математической формулировки условий задачи;

2) наиболее эффективного метода решения задачи;

3) алгоритма, обеспечивающего получение результата с требуемой или возможной точностью при минимальных затратах времени.

Однако на каждом из этих этапов необходимо учитывать особенности применяемого вычислительного средства для наиболее полного их использования.

Результаты инженерных вычислений имеют смысл только в том случае, когда они достаточно достоверны и удовлетворительно описывают свойства физического объекта, который должен быть построен по проекту. Поэтому вопросы точности вычислений, включая оценку требуемой точности исходных данных, при инженерном проектировании имеют первостепенное значение.



Рис. 2. Обобщенная схема инженерного проектирования

2. ТОЧНОСТЬ И ПОГРЕШНОСТИ ВЫЧИСЛЕНИЙ

Численные результаты инженерных расчетов математически моделируют физические величины, истинные значения которых рассматриваются как точные числа. Поэтому мера точности результата вычислений определяется степенью его приближения к значению моделируемой физической величины. Ошибку в описании точного числа a^* его приближенным значением a оценивают абсолютной

$$\varepsilon_a = a^* - a \quad (9a)$$

или относительной (часто выражаемой в процентах)

$$\varepsilon_0 = \varepsilon_a / |a| \quad (9b)$$

погрешностью. Так как точное значение числа не всегда известно, то погрешностями обычно называют (если это не вызывает недоразумений) их предельные значения, определяющие границы интервала, в котором находится точное число.

Результаты инженерных вычислений приближены даже при точном выполнении правил вычислений. Источниками погрешностей результатов являются погрешности исходных данных, представления чисел и метода вычислений.

Погрешности исходных данных принципиально неустранимы вследствие невозможности построения математической модели, адекватной моделируемому физическому объекту. Это ограничение связано как с невозможностью полного учета бесконечного числа физических явлений, определяющих свойства моделируемого объекта, так и с невозможностью абсолютно точного измерения параметров математической модели. Влияние погрешностей параметров x_1, \dots, x_m модели на результат вычислений оценивают по разложению функциональной зависимости, связывающей результат вычислений с исходными значениями параметров, в m -мерный ряд Тейлора

$$f(x_1, \dots, x_m) = f(x_{10}, \dots, x_{m0}) + \sum_{k=1}^m \frac{\partial f_0}{\partial x_k} \delta x_k + \\ + \frac{1}{2} \sum_{k=1}^m \sum_{j=1}^m \frac{\partial^2 f_0}{\partial x_k \partial x_j} \delta x_k \delta x_j + \dots, \quad (10a)$$

где приращения параметров $\delta x_k = x_k - x_{k0}$ относительно точных значений x_{k0} равны абсолютным погрешностям. Для достаточно малых приращений допустимо ограничение ряда линейными членами с вычислением абсолютной погрешности результата

$$\varepsilon_{af} = \delta f \equiv f(x_1, \dots, x_m) - f(x_{10}, \dots, x_{m0}) \approx \sum_{k=1}^m \frac{\partial f_0}{\partial x_k} \delta x_k. \quad (10b)$$

Тогда относительная погрешность результата

$$\epsilon_{of} = \frac{\delta f}{f} = \sum_{k=1}^m \frac{x_k}{f} \frac{\partial f_0}{\partial x_k} \frac{\delta x_k}{x_k} = \sum_{k=1}^m S_f(x_k) \frac{\delta x_k}{x_k}, \quad (10b)$$

где величины $S_f(x_k)$ называют чувствительностями (первого порядка) функции f к изменениям параметров x_k .

Формулы (10) без учета знака производных и приращений пригодны также для оценки предельных абсолютной и относительной погрешностей результата вычислений $|f(x)|$ по предельным абсолютным погрешностям $|\delta x_k|$ исходных данных.

Пример 1. Определить точность вычисления мощности, рассеиваемой на резисторе с сопротивлением $r = 10 \text{ кОм} \pm 5\%$, если к нему приложено напряжение $u = 5 \text{ В}$, измеренное с точностью 1%.

Согласно формуле (10б) для функции $P = u^2/r$ вычисляем $\partial P/\partial u = 2u/r = 10^{-8} \text{ А}$; $\partial P/\partial r = -u^2/r^2 = -25 \cdot 10^{-8} \text{ А}^2$. При $\delta r = 500 \text{ Ом}$ $\delta u = 0,05 \text{ В}$ получаем $\delta P = |10^{-3} \cdot 0,05| + |-25 \cdot 10^{-8} \cdot 500| = 0,175 \text{ мВт}$. Относительная погрешность $\delta P/P = \pm 7\%$ при $P = 25/10^4 = 2,5 \text{ мВт}$.

Расчет чувствительностей позволяет непосредственно оценить влияние погрешностей каждого параметра на погрешность результата. Значения $S_p(u) = 2u^2/r/u^2r = 2$; $S_p(r) = -u^2r^2/u^2r^2 = -1$ свидетельствуют, что погрешность измерения напряжения влияет в большей степени на погрешность вычисления мощности. Относительная погрешность результата вычисляется через чувствительности как $\delta P/P = |2 \cdot 1\%| + |-1 \cdot 5\%| = \pm 7\%$.

Погрешность, вносимую ограничением ряда (10а) линейными членами, в рассматриваемом простом примере легко оценить, вычислив непосредственно предельные значения мощности для наихудшего сочетания знаков приращений параметров:

$$P_{\max} = \frac{u_{\max}^2}{r_{\min}} = \frac{5,05^2}{9,5} = 2,6844736 \text{ мВт}; P_{\min} = \frac{u_{\min}^2}{r_{\max}} = \frac{4,95^2}{10,5} = 2,3335714 \text{ мВт}.$$

Следовательно, $-6,66\% < \delta P/P < 7,38\%$, т. е. использование формул (10в) даёт удовлетворительную оценку.

Оценка погрешности результата по предельным значениям погрешностей исходных данных обычно приводит к завышенным значениям, особенно при большом числе некоррелированных параметров. Поэтому часто прибегают к среднеквадратичному суммированию

$$\frac{\delta f}{f} \approx \left[\sum_{k=1}^m \left[S_f(x_k) \frac{\delta x_k}{x_k} \right]^2 \right]^{1/2}.$$

Эта оценка более близка к истинному значению, так как вероятность наихудшего сочетания параметров, приводящего к максимальным значениям погрешности, вычисляемым по формуле (10в), при достаточно большом числе m становится пренебрежимо малой.

Точность инженерных расчетов повышается при учете большего числа сторонних воздействий, но в этом случае математическая модель усложняется, а требуемое время вычислений увеличивается.

Поэтому необходимый верхний предел сложности математической модели для заданных рабочих условий зависит от требуемой точности результатов.

Пример 2. Определить погрешность вычисленных в предыдущем примере значений мощности на частотах $f_1 = 500$ кГц и $f_2 = 500$ МГц, если паразитная индуктивность резистора $L = 1$ мкГн при действующем напряжении $u = 5$ В.

С учетом паразитной индуктивности рассеиваемая на резисторе мощность вычисляется по формуле

$$P = u^2 r / [r^2 + (2\pi f)^2 L^2].$$

Следовательно, при $f_1 = 500$ кГц

$$P = \frac{25 \cdot 10^4}{10^8 + 4\pi^2 25 \cdot 10^{10} \cdot 10^{-12}} = 2,4999997 \text{ мВт},$$

а при $f_2 = 500$ МГц

$$P = \frac{25 \cdot 10^4}{10^8 + 4\pi^2 25 \cdot 10^{16} \cdot 10^{-12}} = 2,275424 \text{ мВт}.$$

Таким образом, если на частоте f_1 погрешность результата вычислений в примере 1 пренебрежимо мала ($\delta P/P \approx 0,000012\%$), то на частоте f_2 эта погрешность $\delta P/P \approx 9\%$ превышает влияние погрешностей определения u и r .

При десятичном счислении число отображается последовательностью разрядов, в каждый из которых записывается десятичный знак (цифра). Точность приближенного числа часто оценивают по количеству верных цифр в его десятичном представлении. Цифры называют верными, если абсолютная погрешность числа не превышает половины единицы последнего разряда. Например, точное значение числа 32,075 со всеми верными цифрами находится в пределах от 32,0745 до 32,0755. Если дробная часть числа с верными цифрами оканчивается нулями, то их сохраняют для указания на точность числа. Например, десятичное представление числа 435,800 означает, что это число определено с точностью до шести верных значащих* цифр и его точное значение заключено в пределах от 435,7995 до 435,8005.

Представление оканчивающихся нулями целых чисел в естественной (с фиксированной запятой) форме не позволяет непосредственно судить о их точности. Так, по виду числа 14 000 нельзя определить число верных цифр. Более удобно представление чисел с плавающей запятой $a \cdot 10^n$, где a — число с запятой, фиксированной перед первой значащей цифрой или после определенного их количества (разрядов), иногда называемое мантиссой, а n — целое число, называемое порядком. В стандартной форме такого представления запятая в мантиссе фиксируется после первой значащей цифры. В этом случае для указания на три верных цифры числа 14 000 оно записывается в стандартной форме $1,40 \cdot 10^4$. В ЭВМ числа обычно представляют в нормализованной форме, когда запятая фиксируется перед первой значащей цифрой (например, $0,140 \cdot 10^5$).

* Значащими называют все цифры десятичного представления числа от 1 до 9, а также нули, расположенные между ними и, при указании точности, справа от них.

Верные цифры не обязательно совпадают с цифрами точного числа. Например, для точного числа 4 в его приближении 3,9999 первые четыре цифры верные, так как оно приближает все числа, лежащие в пределах от 3,9994 до 4,0004.

Количество верных цифр служит мерой как абсолютной, так и относительной погрешности приближенных чисел. Если число представлено k ($k \geq 2$) верными цифрами, то его относительная погрешность

$$\epsilon_0 \leqslant 1/(2a_m 10^{k-1}), \quad (11a)$$

где a_m — первая значащая цифра. При $k = 1$

$$\epsilon_0 \leqslant 1/a_m. \quad (11b)$$

По формулам (11) определяют также количество верных цифр, необходимых для представления числа с заданной точностью.

Приближенное число содержит n верных значащих цифр в широком смысле, если его абсолютная погрешность не превосходит единицы n -го разряда. Например, для числа 32,075 его приближения 32,076 и 32,074 содержат четыре верных (в узком смысле) цифры и пять верных в широком смысле цифр. Если абсолютная погрешность числа не превосходит двух единиц последнего разряда, то содержащуюся в нем цифру называют сомнительной.

Исходные данные инженерных расчетов не всегда содержат все верные цифры. Поэтому перед началом расчета в этих числах следует сохранить лишь верные цифры, что не увеличит погрешность результатов вычислений, но сократит затраты времени. Более того, для сокращения времени вычислений часто используют представления чисел лишь с таким количеством значащих цифр (разрядов), которое обеспечивает требуемую точность результата.

Простейший способ округления десятичного представления числа до k разрядов заключается в сохранении первых k и отбрасывании остальных значащих цифр (с заменой их нулями при округлении целых чисел). Относительная погрешность округления в этом случае определяется формулами (11), но последняя сохраняемая цифра оказывается верной лишь в широком смысле, если первая отбрасываемая верная цифра больше 5.

Погрешность уменьшается при «симметричном» округлении, когда последняя сохраняемая цифра увеличивается на единицу, если первая отбрасываемая верная цифра больше 5. Более точно «несимметричное» округление, отличающееся от симметричного тем, что при нечетности последней сохраняемой цифры она увеличивается на единицу, если первая отбрасываемая цифра равна 5.

Во многих случаях для получения точного результата в операндах* нужно сохранять чрезмерно большое (иногда бесконечное) количество верных цифр. Так как такие операнды и результаты опе-

* Операндом называют число, над которым производится математическая операция.

раций неизбежно приходится округлять, то даже при использовании точных математических моделей возникают погрешности результата, связанные с погрешностями округления операндов. При последующих вычислениях эти погрешности накапливаются («распространение ошибки») и при большом числе операций погрешности округления могут существенно снизить точность результатов. Этую погрешность, часто называемую операционной, можно уменьшить, выбрав другой алгоритм вычислений с меньшим числом операций.

Операционные погрешности оценивают по формулам (10), подразумевая под переменными x_k округляемые операнды, а под приращениями δx_k — их абсолютные погрешности округления. Вместо соотношений (10) удобно использовать граф вычислений, наглядно иллюстрирующий влияние каждой погрешности на результаты промежуточных и окончательного результата. Вершины такого графа отображают исходные операнды и результаты каждой операции, а веса его ветвей равны чувствительности результатов соответствующих операций к изменениям операндов. При определении предельных погрешностей можно учитывать лишь модули операционных чувствительностей:

$$\begin{aligned} |S_{a \pm b}(a)| &= |a/(a \pm b)|; \quad |S_{a \pm b}(b)| = |b/(a \pm b)|; \\ |S_{ab}(a)| &= |S_{ab}(b)| = |S_{a/b}(a)| = |S_{a/b}(b)| = 1; \\ |S_{a^m}(a)| &= m; \quad |S_{\sqrt{a}}(a)| = 0,5 \text{ и т. д.} \end{aligned}$$

Относительные погрешности округления операндов и результатов операций можно указывать на таком графике стрелками, направленными к соответствующей вершине. Влияние погрешности операнда на любой результат определяется произведением этой погрешности на веса дуг, соединяющих ее с рассматриваемым результатом, которое численно равно относительной операционной погрешности.

Пример 3. Оценить предельную погрешность результата вычислений функции $y = (\sqrt{a} + \sqrt{b})^2$ при несимметричном округлении всех чисел до трех знаков после запятой (с абсолютной погрешностью округления $0,5 \cdot 10^{-3}$) при исходных данных $a = 2,084$; $b = 9,744$.

При округлении результатов извлечения корней сложение не вносит погрешностей. Поэтому составляющие операционной погрешности в соответствии с формулой (10б)

$$\begin{aligned} \delta y_1 &= \delta(\sqrt{a})dy/d(\sqrt{a}) = 2(\sqrt{a} + \sqrt{b}) \cdot 0,5 \cdot 10^{-3} \approx 4,6 \cdot 10^{-3}; \\ \delta y_2 &= \delta(\sqrt{b})dy/d(\sqrt{b}) = 2(\sqrt{a} + \sqrt{b}) \cdot 0,5 \cdot 10^{-3} \approx 4,6 \cdot 10^{-3}; \\ \delta y_3 &= \delta((\sqrt{a} + \sqrt{b})^2)dy/dy = 0,5 \cdot 10^{-3}, \end{aligned}$$

и предельная абсолютная погрешность результата

$$\delta y_{\max} = \delta y_1 + \delta y_2 + \delta y_3 \approx 9,7 \cdot 10^{-3}$$

при истинном значении этой погрешности $\delta y \approx 7,8 \cdot 10^{-3}$.

Если изменить алгоритм вычислений, преобразовав исходное выражение к виду $y = a + b + 2\sqrt{ab}$, то составляющие операционной погрешности

$$\begin{aligned} \delta y_1 &= \delta(ab)dy/d(ab) = 0,5 \cdot 10^{-3}/\sqrt{ab} \approx 0,11 \cdot 10^{-3}; \\ \delta y_2 &= \delta(\sqrt{ab})dy/d(\sqrt{ab}) = 10^{-3}; \\ \delta y_3 &= \delta(2\sqrt{ab})dy/d(2\sqrt{ab}) = 0,5 \cdot 10^{-3}. \end{aligned}$$

Следовательно, в этом случае предельная погрешность $\delta y_{\max} \approx 1,61 \cdot 10^{-3}$ при истинном значении абсолютной погрешности $\delta y \approx 0,54 \cdot 10^{-3}$.

По графу вычислений согласно формуле $y = (\sqrt{a} + \sqrt{b})^2$, представленному на рис. 3, а, находим

$$\frac{\delta y_{\max}}{y} = \left(\frac{0,5 \cdot 10^{-3}}{\sqrt{a}} \cdot \frac{\sqrt{a}}{\sqrt{a} + \sqrt{b}} + \frac{0,5 \cdot 10^{-3}}{\sqrt{b}} \cdot \frac{\sqrt{b}}{\sqrt{a} + \sqrt{b}} \right) \cdot 2 + \\ + \frac{0,5 \cdot 10^{-3}}{(\sqrt{a} + \sqrt{b})^2} \approx 0,462 \cdot 10^{-3}$$

или $\delta y_{\max} \approx 9,63 \cdot 10^{-3}$.

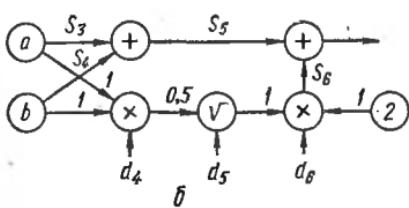
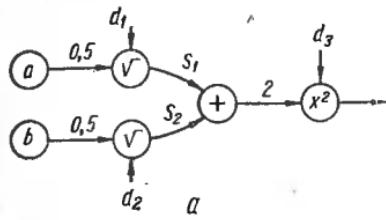


Рис. 3. Графы вычислений.

$$S_1 = \sqrt{a}/(\sqrt{a} + \sqrt{b}); S_2 = \sqrt{b}/(\sqrt{a} + \sqrt{b}); S_3 = a/(a+b); S_4 = b/(a+b); S_5 = (a+b)/(a+b+2\sqrt{ab}); S_6 = 2\sqrt{ab}/(a+b+2\sqrt{ab}); d_1 = 0,5 \cdot 10^{-3}/\sqrt{a}; d_2 = 0,5 \times 10^{-3}/\sqrt{b}; d_3 = 0,5 \cdot 10^{-3}/(\sqrt{a} + \sqrt{b})^2; d_4 = 0,5 \cdot 10^{-3}/ab; d_5 = 0,5 \cdot 10^{-3}/\sqrt{ab}; d_6 = 0,5 \cdot 10^{-3}/2\sqrt{ab}$$

По графу вычислений согласно формуле $y = a + b + 2\sqrt{ab}$, представленному на рис. 3, б, находим

$$\frac{\delta y_{\max}}{y} = 0,5 \cdot 10^{-3} \left(\frac{1}{ab} \cdot \frac{1}{2} + \frac{1}{\sqrt{ab}} + \frac{1}{2\sqrt{ab}} \right) \frac{2\sqrt{ab}}{a+b+2\sqrt{ab}} \approx 0,077 \cdot 10^{-3}$$

или $\delta y_{\max} \approx 1,61 \cdot 10^{-3}$.

Максимальная операционная погрешность возникает при вычитании близких чисел, так как в этом случае предельная относительная погрешность разности $a - b$ близких чисел

$$\frac{\delta y_{\max}}{y} = |S_{a-b}(a)| \frac{\delta a}{a} + |S_{a-b}(b)| \frac{\delta b}{b} = \frac{\delta a + \delta b}{|a - b|}$$

может превысить единицу, и все цифры результата окажутся ошибочными.

Исследование путей распространения ошибок при сложных вычислениях достаточно громоздко и в этих случаях используют простые правила [8] округления результатов операций, позволяющие контролировать число верных цифр:

1. В результате сложения и вычитания сохраняется младший разряд, равный наибольшему среди разрядов последних значащих цифр операндов.

2. В результате умножения и деления сохраняется число значащих цифр, равное наименьшему числу значащих цифр операндов.

3. В результате возвведения в квадрат и куб и извлечения квадратного и кубического корней сохраняется число значащих цифр, равное числу значащих верных цифр операнда.

4. Во всех промежуточных результатах сохраняется на одну цифру больше требуемых приведенными правилами, но эта цифра отбрасывается в окончательном результате.

5. Если допустимо выбирать точность operandов, то их следует брать с таким числом верных цифр, которое по приведенным оценкам позволяет получить в результате число верных цифр, на единицу большее требуемого.

Существенное практическое значение имеет обратная задача исследования погрешностей — определение допустимых погрешностей operandов при заданной предельной погрешности результата вычислений. В простейшем случае решение этой задачи основано на предположении о равных вкладах m составляющих правой части формулы (10б) в погрешность результата. При этом предположении для заданной предельной абсолютной погрешности ϵ_{af} результата допустимые погрешности operandов

$$\epsilon_{ak} \leq \epsilon_{af}/(m |\partial f/\partial x_k|); \quad k = 1, 2, \dots, m. \quad (12a)$$

Пусть, например, необходимо определить допустимые погрешности operandов по допустимой погрешности результата 0,1 мВт при решении задачи по условиям примера 1. По вычисленным в примере значениям производных согласно формуле (12a) при $m = 2$ находим допустимые абсолютные погрешности исходных данных $\epsilon_u = 0,05$ В и $\epsilon_r = 200$ Ом и соответствующие допустимые относительные погрешности $\epsilon_{ou} = 1\%$ и $\epsilon_{or} = 2\%$.

В предположении о равенстве вкладов в соответствии с формулой (12a) для некоторых operandов можно получить практически невыполнимые допустимые погрешности. Поэтому во многих случаях целесообразно использовать допущение о равенстве предельных абсолютных погрешностей $\epsilon_1 = \dots = \epsilon_m$ operandов одинаковой размерности. Тогда для заданной предельной погрешности результата в соответствии с формулой (10б) допустимые значения абсолютных погрешностей operandов

$$\epsilon_{ak} \leq \epsilon_{af} / \sum_{j=1}^m |\partial f/\partial x_j|; \quad k = 1, 2, \dots, m. \quad (12b)$$

Принятое при выводе этой формулы допущение часто вытекает непосредственно из условий задачи.

Пример 4. Определить допуски на длины сторон $a = 10$ см, $b = 20$ см, $c = 30$ см, если допустимая погрешность объема параллелепипеда с такими сторонами не должна превышать 10 см^3 .

Вычисляем $\partial abc/da = 600$ см, $\partial abc/db = 300$ см, $\partial abc/dc = 200 \text{ см}^2$ и в соответствии с формулой (12b) находим допуск на длины сторон $\epsilon \leq 10/1100 = 0,0091$ см.

Большой общностью характеризуется предположение о равенстве предельных относительных погрешностей операндов. В этом случае согласно формуле (10в) допустимые абсолютные погрешности операндов

$$\varepsilon_{ak} \leq |x_k| \varepsilon_{af} / \sum_{j=1}^{n_e} |x_j \partial f / \partial x_j|; \quad k = 1, 2, \dots, m. \quad (12\text{в})$$

Практически достижимая точность чисел, описывающих физические величины, ограничена возможной точностью их измерения или технологией изготовления физических объектов, свойства которых моделируются этими числами. Поэтому для некоторых операндов требуемая согласно формул (12) точность может оказаться практически недостижимой, тогда как другие операнды могут определяться более точно, чем требуется по этим формулам. В таких случаях целесообразно разбивать операнды на группы в соответствии с возможной точностью их определения и так подбирать для каждой группы формулы (12), чтобы наилучшим образом удовлетворить практические ограничения на точность операндов.

Результаты вычислений даже при отсутствии операционных погрешностей и точных моделях иногда отличаются низкой точностью, связанной с погрешностями выбранного метода решения задачи. Методы численного решения разбиваются на точные (прямые) и приближенные (последовательных приближений). Точные методы основаны на выполнении заранее известного конечного числа операций, приводящих при отсутствии погрешностей исходных данных и погрешностей округления к точному значению результата. В приближенных методах выбранное начальное значение результата последовательно приближается к его точному значению в теоретически бесконечном процессе вычислений. Так как практически выполняется лишь конечное число операций, то при этом возникает методическая ошибка, называемая остаточной погрешностью, которая уменьшается при увеличении числа приближений.

Многие методы последовательных приближений нечувствительны к операционным погрешностям и случайным ошибкам, которые увеличивают время вычислений, но не влияют на конечный результат. Однако в некоторых приближенных методах по мере уменьшения остаточной погрешности возрастает операционная ошибка, вследствие чего результат можно вычислить лишь с ограниченной точностью.

Пример 5. Оценить погрешность вычисления числа π по периметрам вписанного в окружность и описанного вокруг нее правильных m -угольников при бесконечном удвоении их сторон и округлении операндов отбрасыванием до восьми значащих цифр.

Стороны a_0 вписанного и b_0 описанного шестиугольников (рис. 4) связаны с радиусом окружности соотношениями $a_0 = R$, $b_0 = 2R/\sqrt{3}$. Начальное приближение числа π в этом случае определяется неравенством $6a_0/2R < \pi_0 < 6b_0/2R$ или $3 < \pi_0 < 2\sqrt{3}$.

При $R = 1$ и i -м удвоении сторон многоугольника

$$a_i = \sqrt{2 - \sqrt{4 - a_{i-1}^2}}; b_i = (2\sqrt{4 + b_{i-1}^2} - 4)/b_{i-1}; 3 \cdot 2^i a_i < \pi_i < 3 \cdot 2^i b_i.$$

Операционную погрешность вычисления a_i и b_i на каждом i -м удвоении сторон можно определить по графикам, показанным на рис. 5. Для оценки асимптотических значений ошибки при $i > 1$ используем приближения: $a_{i-1} \approx 1/2^{i-1}$; $b_{i-1} \approx 1/2^{i-1}$; $4 - a_{i-1}^2 \approx 4$; $4 + b_{i-1}^2 \approx 4$; $a_{i-1}^2/(4 - a_{i-1}^2) \approx 1/2^{2i}$; $2 - \sqrt{4 - a_{i-1}^2} \approx 2 - 2(1 - 0,125a_{i-1}^2) \approx 0,25a_{i-1}^2 \approx 1/2^{2i}$; $2\sqrt{4 + b_{i-1}^2} - 4 \approx$

$$\approx 4(1 - 0,125b_{i-1}^2) - 4 \approx 0,5b_{i-1}^2 \approx 1/2^{2i}.$$

Тогда

$$\frac{\delta a_i}{a_i} \approx 10^{-7} \left(2^i + \frac{2^{2i}}{2} + \frac{2^{2i}}{2} + \frac{2^{2i}}{8} + \frac{2^{2i}}{8} \right) + \\ + \frac{\delta a_{i-1}}{a_{i-1}} = 2i \cdot 10^{-7} (1,25 \cdot 2i + 1) + \\ + \delta a_{i-1}/a_{i-1} \approx 1,25 \cdot 2^{2i} \cdot 10^{-7}$$

и абсолютная операционная погрешность при приближении окружности вписанным многоугольником $\delta \pi_{\text{вп } i} \leq 3,75 \cdot 2^{2i} \cdot 10^{-7}$.

Аналогично находим для описанного многоугольника

$$8b_i/b_i = 10^{-7} \cdot 2^i (1 + 2,5 \cdot 2^i) + \\ + 2\delta b_{i-1}/b_{i-1} \approx 2,5 \cdot 2^{2i} \cdot 10^{-7} \text{ и} \\ \delta \pi_{\text{оп } i} \leq 7,5 \cdot 2^{2i} \cdot 10^{-7}.$$

Рис. 4. Вписанный в окружность и описанный вокруг нее шестиугольники

Сравнивая значения $(a_i - b_i)$ и $(a_{i-1} - b_{i-1})$, получаем значение методической погрешности $\delta \pi_i \approx 0,5 \cdot 2^{-2i}$.

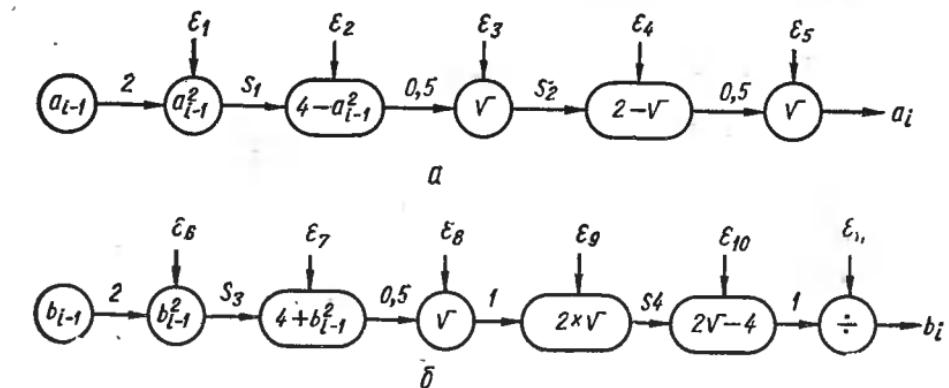


Рис. 5. Графы вычислений:

$$S_1 = a_{i-1}^2/(4 - a_{i-1}^2); S_2 = \sqrt{4 - a_{i-1}^2}/(2 - \sqrt{4 - a_{i-1}^2}); S_3 = b_{i-1}^2/(4 + b_{i-1}^2); \\ S_4 = 2\sqrt{4 + b_{i-1}^2}/(2\sqrt{4 + b_{i-1}^2} - 4); \epsilon_k = 10^{-7}/y_k, \text{ где } y_k \text{ — результат } k\text{-й операции}$$

Суммарные погрешности вычислений

$$\delta\pi_{\text{вп}} \approx 0,5 \cdot 2^{-2i} + 3,75 \cdot 2^{2i} \cdot 10^{-7};$$

$$\delta\pi_{\text{оп}} \approx 0,5 \cdot 2^{-2i} + 7,5 \cdot 2^{2i} \cdot 10^{-7}.$$

Подстановкой определяем значение $i = 5$, при котором минимальны суммарные погрешности

$$\delta\pi_{\text{вп}} \leq 0,00049 + 0,00038 \approx 0,00087;$$

$$\delta\pi_{\text{оп}} \leq 0,00049 + 0,00077 \approx 0,00126.$$

Следовательно, при выбранном методе вычислений и округлении операндов до восьми знаков отбрасыванием нельзя получить значение π более чем с тремя верными цифрами.

Если выбранный приближенный метод не обеспечивает требуемую точность результата вычисления вследствие операционной погрешности, то следует выбрать другой метод или алгоритм вычислений с большей скоростью сходимости процесса последовательных приближений.

3. МОДЕЛИРОВАНИЕ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТА

Математические модели физических зависимостей вида $y(x)$ составляют по результатам одновременного измерения величин x и $y = y(x)$ в интервале изменения аргумента x , определяемого заданными рабочими условиями.

Полученные в результате одного измерения значения x и $y = y(x)$ являются случайными вследствие погрешностей измерения и влияния сторонних воздействий. В практических задачах обычно ограничиваются определением таких числовых характеристик этих случайных величин как математическое ожидание (среднее) или первый (начальный) момент $m_1(x)$ и $m_1(y)$ и дисперсия или второй (центральный) момент $M_2(x) = m_2[x - m_1(x)] = m_2(x) - m_1^2(x)$; $M_2(y) = m_2(y) - m_1^2(y)$, описывающих соответственно наиболее вероятное значение случайной величины и вероятную степень отклонения от этого значения. Мерой отклонения может служить среднеквадратичное отклонение $\sigma = \sqrt{M_2}$.

Если все N измерений случайного числа y выполнены с одинаковой точностью, то эмпирические моменты k -го порядка оценивают по формулам вычисления начальных

$$\tilde{m}_k(y) = \frac{1}{N} \sum_{j=1}^N y_j^k; \quad j = 1, 2, \dots, N \quad (13a)$$

и центральных

$$\widetilde{M}_k(y) = \frac{1}{N} \sum_{j=1}^N [y_j - m_1(y)]^k; \quad j = 1, 2, \dots, N \quad (13b)$$

моментов.

Оценка дисперсии случайной величины, распределенной по нормальному закону, чаще всего встречающемуся на практике, при вычислении по формуле (13€) при $k = 2$ оказывается смещенной. Поэтому в этом случае полагают $\tilde{\sigma}^2 = \bar{M}_2(y)N/(N - 1)$.

Так как полученные оценки сами являются случайными величинами, то статистическую обработку результатов эксперимента дополняют анализом возможных погрешностей этих оценок. Например, можно грубо принять, что среднеквадратичная погрешность эмпирического среднего \tilde{m}_1 (его вероятное отклонение от истинного значения среднего m_1) $\tilde{\sigma}_{m_1} = \tilde{\sigma}/\sqrt{N}$ или, на основании правила «трех сигм», эмпирическое среднее отклоняется от истинного с доверительной вероятностью около 0,997 не более $|\tilde{m}_1 - m_1| \leq 3\tilde{\sigma}/\sqrt{N}$.

Более точная оценка интервала, в котором с заданной доверительной вероятностью $P_{\text{дов}}$ находится истинное значение среднего (доверительный интервал), при нормальном распределении ошибок требует определения коэффициента t , зависящего [21] от $P_{\text{дов}}$ и числа N измерений (табл. 1). С учетом этого коэффициента

$$|\tilde{m}_1 - m_1| < t\tilde{\sigma}/\sqrt{N}. \quad (13\text{в})$$

Таблица 1

$P_{\text{дов}}$	0,90	0,95	0,98	0,99	0,999	$P_{\text{дов}}$	0,90	0,95	0,98	0,99	0,999
N	t					N	t				
4	2,132	2,776	3,747	4,604	8,610	20	1,725	2,086	2,528	2,845	3,850
5	2,015	2,571	3,365	4,032	6,859	25	1,708	2,060	2,485	2,787	3,725
6	1,943	2,447	3,143	3,707	5,959	30	1,697	2,042	2,457	2,750	3,646
7	1,895	2,365	2,998	3,499	5,405	35	1,689	2,030	2,437	2,724	3,591
8	1,860	2,306	2,896	3,355	5,041	40	1,684	2,021	2,423	2,704	3,551
9	1,833	2,262	2,821	3,250	4,781	45	1,679	2,014	2,412	2,689	3,522
10	1,812	2,228	2,764	3,169	4,587	50	1,676	2,008	2,403	2,677	3,497
11	1,796	2,201	2,718	3,106	4,487	60	1,671	2,000	2,390	2,660	3,460
12	1,782	2,179	2,681	3,055	4,318	70	1,667	1,995	2,381	2,648	3,436
13	1,771	2,160	2,650	3,012	4,221	80	1,664	1,990	2,374	2,639	3,416
14	1,761	2,145	2,624	2,977	4,140	90	1,662	1,987	2,368	2,632	3,401
15	1,753	2,131	2,602	2,947	4,073	100	1,660	1,984	2,364	2,626	3,391
16	1,746	2,120	2,583	2,921	4,015	∞	1,645	1,960	2,326	2,576	3,291
18	1,734	2,103	2,552	2,878	3,922						

С помощью коэффициента q , значения которого для различных значений $P_{\text{дов}}$ и $N - 1$ приведены в табл. 2 [21], можно найти доверительный интервал для полученной оценки среднеквадратичного отклонения

$$|\tilde{\sigma} - \sigma| < q\tilde{\sigma} \quad (13\text{г})$$

в предположении о нормальном законе распределения отклонений.

Так как правильность найденных оценок зависит от действительного закона распределения случайной величины, то иногда воз-

Таблица 2

$P_{\text{дов}}$	0,95	0,99	0,999	$P_{\text{дов}}$	0,95	0,99	0,999
N	q			N	q		
17	0,400	0,639	0,961	55	0,196	0,280	0,398
18	0,385	0,602	0,916	60	0,187	0,266	0,375
19	0,371	0,578	0,875	65	0,179	0,253	0,356
20	0,358	0,556	0,838	70	0,172	0,242	0,339
22	0,336	0,518	0,776	80	0,160	0,224	0,311
24	0,318	0,487	0,724	90	0,150	0,209	0,289
26	0,302	0,460	0,679	100	0,142	0,196	0,271
28	0,288	0,437	0,641	110	0,135	0,186	0,257
30	0,276	0,416	0,609	120	0,129	0,177	0,246
35	0,253	0,375	0,544	150	0,115	0,159	0,220
40	0,234	0,343	0,494	200	0,099	0,135	0,184
45	0,219	0,318	0,455	250	0,089	0,120	0,162
50	0,207	0,297	0,424	300	0,081	0,109	0,146

никает необходимость в определении этого закона или, по крайней мере, в проверке нормальности распределения. Подобную проверку можно выполнить по эмпирическим центральным моментам третьего и четвертого порядков, определив с их помощью коэффициенты асимметрии и эксцесса

$$\tilde{\gamma}_3 = \widetilde{M}_3 / \sqrt{\widetilde{M}_2^3}, \quad \tilde{\gamma}_4 = (\widetilde{M}_4 / \widetilde{M}_2^2) - 3.$$

Для нормального закона распределения эти коэффициенты равны нулю. Поэтому, если вычисленные значения γ_3 и γ_4 в 2—3 раза превышают величины

$$\sqrt{\frac{6(N-1)}{(N+1)(N+3)}}; \quad \sqrt{\frac{24N(N-2)(N-3)}{(N-1)^2(N+3)(N+5)}} \quad (13d)$$

соответственно, то нормальность закона распределения сомнительна.

При совместной статистической обработке результатов измерения двух случайных величин x и y может возникнуть необходимость в определении эмпирического коэффициента корреляции между ними

$$\tilde{r} = \frac{\sum_{j=1}^N [y_j - \tilde{m}_1(y)][x_j - \tilde{m}_1(x)]}{(N-1)\tilde{\sigma}_x\tilde{\sigma}_y}. \quad (13e)$$

Доверительные оценки коэффициента корреляции сложны и разработаны главным образом для нормального совместного распределения x и y . Приближенную оценку доверительного интервала \tilde{r} можно определить по номограммам, приведенным в работе [1].

Вместо определения доверительного интервала коэффициента корреляции часто ограничиваются проверкой значимости коэф-

Таблица 3

$P_{\text{дов}}$	0,9	0,95	0,99	0,999	$P_{\text{дов}}$	0,95	0,99	0,999	
N	H				N	H			
10	1,65	1,90	2,29	2,62	25	1,941	2,475	3,026	
11	1,65	1,90	2,32	2,68	26	1,941	2,479	3,037	
12	1,65	1,91	2,35	2,73	27	1,942	2,483	3,047	
13	1,65	1,92	2,37	2,77	28	1,943	2,487	3,056	
14	1,65	1,92	2,39	2,81	29	1,943	2,490	3,064	
15	1,65	1,92	2,40	2,85	30	1,944	2,492	3,071	
16	1,65	1,93	2,41	2,87	35	1,947	2,505	3,102	
17	1,65	1,93	2,42	2,90	40	1,949	2,514	3,126	
18	1,65	1,93	2,43	2,92	45	1,950	2,521	3,145	
19	1,65	1,93	2,44	2,94	50	1,951	2,527	3,161	
20	1,65	1,94	2,45	2,96	60	1,953	2,535	3,183	
21	1,65	1,94	2,45	2,98	70	1,954	2,541	3,188	
22	1,65	1,94	2,46	2,99	80	1,955	2,546	3,209	
23	1,65	1,94	2,47	3,00	100	1,956	2,553	3,226	
24	1,65	1,94	2,47	3,02	∞	1,960	2,576	3,291	

фициента корреляции, т. е. проверкой допустимости отказа от гипотезы о корреляции между рассматриваемыми величинами. Для проверки сравнивают значение $|\tilde{r}| \sqrt{N - 1}$ с критическим значением H этого произведения (табл. 3) при заданной надежности $P_{\text{дов}}$ выхода [21].

Эмпирическое значение нормированной автокорреляционной функции величины $y(x)$ вычисляют по формуле

$$\tilde{r}(S \Delta x) = \frac{\sum_{i=1}^{N-S} y_i y_{i+S} - (N-S) \left(\sum_{i=1}^N y_i \right)^2 / N^2}{\left[\sum_{i=1}^N y_i^2 - \left(\sum_{i=1}^N y_i \right)^2 / N \right] (N-S-1)/(N-1)}, \quad (13ж)$$

где $S \Delta x$ — интервал между двумя значениями $y(x_i)$ при шаге Δx аргумента. Надежность результатов вычислений по этой формуле оценивают аналогично доверительному интервалу коэффициента корреляции.

При отличии закона распределения от нормального характер изучаемого распределения можно грубо оценить по эмпирическим моментам. В табл. 4 приведены характеристики основных типов распределений, встречающихся в инженерной практике. Следует учитывать, что точность определения эмпирических моментов высоких порядков мала и такая оценка имеет малую надежность.

При статистических исследованиях иногда возникает необходимость в моделировании случайной величины с заданным распределением. Несложный алгоритм, позволяющий вычислять случайные числа с равномерным законом распределения в интервале $[-1/2; 1/2]$, описывается формулой [6]

$$x_i = (x_{i-1} - x_{i-k})_{\text{mod } 1}, \quad (14a)$$

Таблица 4

Законы распределения	m_1	M_2	γ_3	γ_4
Дискретные:				
равновероятный				
$P(\xi = k) = \begin{cases} 0 & \xi < 1 \\ \frac{1}{m} & 1 \leq \xi \leq m \\ 0 & \xi > m \end{cases}$	$\frac{m+1}{2}$	$\frac{m^2 - 1}{12}$	0	$-1,2 + \frac{4}{m^2 - 1}$
биномиальный				
$P(\xi = k) = \begin{cases} 0 & \xi < 0 \\ C_m^k p^k (1-p)^{m-k} & 0 \leq \xi \leq m \\ 0 & \xi > m \end{cases}$	mp	$\frac{mpq}{(q=1-p)}$	$\frac{1-2p}{\sqrt{mpq}}$	$\frac{1-6pq}{mpq}$
Пуассона				
$P(\xi = k) = \begin{cases} 0 & \xi < 0 \\ \frac{\lambda^k}{k!} e^{-\lambda} & \xi \geq 0 \end{cases}$	λ	λ	$\frac{1}{\sqrt{\lambda}}$	$\frac{1}{\lambda}$
Непрерывные:				
равномерный				
$p(x) = \begin{cases} 0 & x < a \\ \frac{1}{(b-a)} & a \leq x \leq b \\ 0 & x > b \end{cases}$	$\frac{a+b}{2}$	$\frac{(a-b)^2}{12}$	0	-1,2
Симпсона				
$p(x) = \begin{cases} 0 & x < a \\ \frac{4(x-a)}{(b-a)^2} & a \leq x \leq \frac{a+b}{2} \\ \frac{4(b-x)}{(b-a)^2} & \frac{a+b}{2} \leq x \leq b \\ 0 & x > b \end{cases}$	$\frac{a+b}{2}$	$\frac{(a-b)^2}{24}$	0	-0,6
нормальный				
$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$	μ	σ^2	0	0
Стьюдента				
$p(x) = \frac{1}{\sqrt{\pi m}} \frac{\Gamma\left(\frac{m+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right)} \left(1 + \frac{x^2}{m}\right)^{-\frac{m+1}{2}}$	0	$\frac{m}{m-2}$ ($m \geq 3$)	0 ($m \geq 4$)	$\frac{6}{m-4}$ ($m \geq 5$)
Гамма-распределение				
$p(x) = \begin{cases} 0 & x < 0 \\ \frac{x^{\alpha} e^{-x/\beta}}{\beta^{\alpha+1} \Gamma(\alpha+1)} & x \geq 0 \end{cases}$	$(\alpha+1)\beta$	$(\alpha+1)\beta^2$	$\frac{2}{\sqrt{\alpha+1}}$	$\frac{6}{\alpha+1}$
арксинуса				
$p(x) = \begin{cases} 0 & x < -a \\ \frac{1}{\pi \sqrt{a^2 - x^2}} & -a \leq x \leq a \\ 0 & x > a \end{cases}$	0	$\frac{a^2}{2}$	0	-1,5
Релея				
$p(x) = \begin{cases} \frac{x}{\sigma^2} e^{-x^2/2\sigma^2} & x \geq 0 \\ 0 & x < 0 \end{cases}$	$\sqrt{\frac{\pi}{2}} \sigma$	$\frac{4-\pi}{2} \sigma^2$	0,63	-0,3

где k — умеренно большое число (10—50), а начальные k чисел равномерно распределены в интервале [0; 1].

При неудачном выборе начальных чисел распределение первых вычисленных чисел может существенно отличаться от равномерного, но при достаточно большом числе вычислений распределение приобретает требуемую характеристику.

Во многих случаях удобен метод Коробова [18], согласно которому случайные числа с равномерным распределением вычисляют по формуле

$$x_i = (qx_{i-1})_{\text{mod } p}, \quad (14)$$

где p — большое простое число (2027; 5087; ...), а q подбирают близким к $p/2$ из множества $p = 3^m$ (m — целое число). Интервал [1; $p - 1$] равномерного распределения получаемых по этой формуле чисел x_i делением на p можно преобразовать в интервал [0; 1].

Случайные числа с равномерным распределением в интервале [0; 1] по формуле

$$y_t = \sqrt{2\sigma^2 \ln \frac{1}{x_t}} \quad (14b)$$

преобразуются в случайные числа с распределением Релея, а последующее преобразование

$$z_t = y_t \sin 2\pi x_{t-1} \quad (14c)$$

дает для случайных чисел z_i нормальный закон распределения с той же дисперсией, что и для случайных чисел y_i .

При практической обработке выборки из N отсчетов $y_j(x_i)$ для i -го значения x_i табличной модели удобно оценивать моменты, например среднее

$$y_i = \frac{1}{N} \sum_{j=1}^N \dot{y}_j(x_i) = y_{0i} + \frac{1}{N} \sum_{j=1}^N \delta y_j(x_i), \quad (15)$$

по отклонению δy_j от некоторого фиксированного значения y_{0i} . Значение аргумента x_i при повторных измерениях также определяется с некоторой погрешностью, в связи с чем $y(x_i)$ будет двухмерной случайной величиной, а случайные величины y и x — коррелированными. Однако обычно полагают, что значение x определено достаточно точно и $y(x_i)$ есть одномерная случайная величина.

Наиболее вероятные значения $y(x_i)$ вместе со значениями x_i , $i = 0, 1, \dots, n$ образуют табличную модель исследуемой зависимости. Для удобства последующих вычислений такие модели часто дополняют разностями значений y_i в точках x_i , называемых узлами модели или узлами интерполяции.

При равных расстояниях (шагах) вдоль оси x между узлами $x_0, x_0 + h, \dots, x_0 + nh$ первыми конечными разностями или конечными разностями первого порядка называют числа

$$\Delta y_0 = \Delta y(x_0) = y_1 - y_0 = y(x_0 + h) - y(x_0);$$

$$\Delta y_{n-1} = \Delta y(x_{n-1}) = y_n - y_{n-1} = y(x_0 + nh) - y(x_0 + (n-1)h).$$

Разности этих чисел

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0; \dots; \Delta^2 y_{n-2} = \Delta y_{n-1} - \Delta y_{n-2}$$

называют вторыми конечными разностями и, в общем случае, числа

$$\Delta^m y_t = \Delta^{m-1} y_{t+1} - \Delta^{m-1} y_t \quad (16)$$

называют конечными разностями m -го порядка.

Конечные разности иногда записывают в строках, расположенных между строками вычитаемых чисел, но обычно используют горизонтальные таблицы конечных разностей, подобные показанной в табл. 5.

Таблица 5

t	x_t	y_t	Δy_t	$\Delta^2 y_t$	$\Delta^3 y_t$	$\Delta^4 y_t$	$\Delta^5 y_t$	$\Delta^6 y_t$	$\Delta^7 y_t$	$\Delta^8 y_t$	$\Delta^9 y_t$	$\Delta^{10} y_t$
0	0	7,0	-0,7	-0,4	0,2	-0,5	3,9	-12,4	-2,3	10,2	-10,8	3,4
1	0,1	6,3	-1,1	-0,2	-0,3	3,4	-8,5	-14,7	7,9	-0,6	-7,4	
2	0,2	5,2	-1,3	-0,5	3,1	-5,1	6,2	-6,8	7,3	-8,0		
3	0,3	3,9	-1,8	2,6	-2,0	1,1	-0,6	0,5	-0,7			
4	0,4	2,1	0,8	0,6	-0,9	0,5	-0,1	-0,2				
5	0,5	2,9	1,4	-0,3	-0,4	0,4	-0,3					
6	0,6	4,3	1,1	-0,7	0,0	0,1						
7	0,7	5,4	0,4	-0,7	0,1							
8	0,8	5,8	-0,3	-0,6								
9	0,9	5,5	-0,9									
10	1,0	4,6										

При табличном задании функции ее значения между узлами в общем случае неизвестны и для их вычисления (интерполяции) используют различные гипотезы о характере зависимости $y(x)$. Простейшая основана на предположении о линейной зависимости функций между смежными узлами табличной модели, но в этом случае значения производных в узлах модели неоднозначны, что свидетельствует о неточности подобной кусочно-линейной аппроксимации (приближения) моделируемой зависимости. Поэтому в большинстве случаев интерполяция основана на более сложных предположениях.

При табличном задании зависимости $y_i = y(x_i)$ с постоянным шагом $h = x_{t+1} - x_t$ значения функции между узлами модели опре-

деляют по разностям (16) с помощью многочленов Ньютона для интерполяции вперед

$$P(x) = y_0 + q \Delta y_0 + \frac{q(q-1)}{2!} \Delta^2 y_0 + \dots + \\ + \frac{q(q-1)\dots(q-n+1)}{n!} \Delta^n y_0, \quad (17a)$$

где $q = (x - x_0)/h$, и интерполяции назад

$$P(x) = y_n + q \Delta y_{n-1} + \frac{q(q+1)}{2!} \Delta^2 y_{n-2} + \dots + \\ + \frac{q(q+1)\dots(q+n-1)}{n!} \Delta^n y_0, \quad (17b)$$

где $q = (x - x_n)/h$.

Если в этих формулах принять $n = 1$, то получатся линейные интерполяционные многочлены

$$P(x) = y_0 + q \Delta y_0; \quad P(x) = y_n + q \Delta y_{n-1},$$

а при $n = 2$ — квадратичные (параболические) многочлены

$$P(x) = y_0 + q \Delta y_0 + \frac{q(q-1)}{2} \Delta^2 y_0;$$

$$P(x) = y_n + q \Delta y_{n-1} + \frac{q(q+1)}{2} \Delta^2 y_{n-2}.$$

При построении табличной модели по экспериментальным данным значения аргумента в узлах интерполяции могут оказаться неравноотстоящими, а шаг таблицы переменным. В этих случаях отношения

$$[x_i, x_{i+1}] = \frac{(y_{i+1} - y_i)}{(x_{i+1} - x_i)}; \quad i = 0, 1, \dots, n-1$$

называют разделенными разностями первого порядка, а

$$[x_i, x_{i+1}, x_{i+2}] = \frac{[x_{i+1}, x_{i+2}] - [x_i, x_{i+1}]}{x_{i+2} - x_i}$$

разделенными разностями второго порядка. Разделенные разности m -го порядка вычисляют по разделенным разностям $(m-1)$ -го порядка согласно рекуррентной формулы

$$[x_i, \dots, x_{i+m}] = \frac{[x_{i+1}, \dots, x_{i+m}] - [x_i, \dots, x_{i+m-1}]}{x_{i+m} - x_i}. \quad (18a)$$

Разделенные разности не изменяются при перестановке элементов (например, $[x_0, x_1] = [x_1, x_0]$) и могут записываться в горизонтальную таблицу, подобную табл. 6.

Значение функции между неравноотстоящими узлами модели определяют по разделенным разностям (18a) с помощью формулы Ньютона

$$P(x) = y_0 + [x_0, x_1](x - x_0) + [x_0, x_1, x_2](x - x_0)(x - x_1) + \dots + \\ + [x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \dots (x - x_n). \quad (18b)$$

Таблица 6

i	x_i	y_i	$[x_i, \dots, x_{i+1}]$	$[x_i, \dots, x_{i+2}]$	$[x_i, \dots, x_{i+3}]$	$[x_i, \dots, x_{i+4}]$	$[x_i, \dots, x_{i+5}]$...
0	0	7,0	-10	-20	120,8010	864,7464	-7350,047	...
1	0,25	4,5	-17	31,94444	553,1742	-3545,282	9006,201	...
2	0,35	2,8	-11,25	170,2380	-687,6748	1408,129	-2085,786	
3	0,43	1,9	14,28571	-1,680647	-54,01625	52,36766		
4	0,5	2,9	14	-21,66666	-24,16680			
5	0,6	4,3	7,5	-33,75				
6	0,8	5,8	-6					
7	1,0	4,6						

Для интерполяции табличных моделей с постоянным или переменным шагом используют также схему Эйткина, согласно которой значения функции $y(x)$ между узлами модели вычисляют по формуле линейной интерполяции:

$$y(x) \approx P_{i, i+1} = \frac{1}{x_{i+1} - x_i} \begin{vmatrix} y_i & x_i - x \\ y_{i+1} & x_{i+1} - x \end{vmatrix} = \frac{y_i(x_{i+1} - x) - y_{i+1}(x_i - x)}{x_{i+1} - x_i} \quad (19a)$$

или по интерполяционным многочленам m -го порядка, выражаемым через многочлены $(m-1)$ -го порядка формулой

$$y(x) \approx P_{0, 1, \dots, m} = \frac{1}{x_m - x_0} \begin{vmatrix} P_{0, 1, \dots, m-1} & x_{m-1} - x \\ P_{1, 2, \dots, m} & x_m - x \end{vmatrix}. \quad (19b)$$

Приведенные формулы используют, в частности, для определения значений табулированных элементарных функций между узлами интерполяции.

Значение аргумента x по заданному значению $y(x)$ между узлами табличной модели вычисляют методами обратного интерполирования. Если шаг табличной модели постоянный, то после замены y многочленом (17a)

$$q = \varphi(q) \equiv \frac{y - y_0}{\Delta y_0} - \frac{\Delta^2 y_0}{2 \Delta y_0} q(q-1) - \dots - \frac{\Delta^n y_0}{n! \Delta y_0} q(q-1) \dots (q-n+1). \quad (20)$$

Приняв начальное приближение $q^{(0)} = (y - y_0)/\Delta y_0$, последовательно вычисляют $q^{(k)} = \varphi(q^{(k-1)})$ для $k = 1, 2, \dots$ до совпадения требуемого числа значащих цифр $q^{(k)}$ и $\varphi(q^{(k-1)})$, после чего вычисляют $x = x_0 + q^{(k)}h$.

При задании табличной модели с переменным шагом для обратной интерполяции используют формулу

$$x = \sum_{i=0}^n \frac{(y - y_1)(y - y_2) \dots (y - y_{i-1})(y - y_{i+1}) \dots (y - y_n)}{(y_i - y_1)(y_i - y_2) \dots (y_i - y_{i-1})(y_i - y_{i+1}) \dots (y_i - y_n)} \quad (21a)$$

или формулу с разделенными разностями, вычисленными относительно y ,

$$x = x_0 + [y_0, y_1](y - y_0) + [y_0, y_1, y_2](y - y_0)(y - y_1) + \dots + [y_0, y_1, \dots, y_n](y - y_0)(y - y_1) \dots (y - y_{n-1}). \quad (216)$$

При достаточно малом шаге h табличной модели значения производных в интервалах между смежными узлами грубо определяют по формулам

$$y'(x) = \frac{\Delta y_i}{h}; \quad y''(x) = \frac{\Delta^2 y_i}{h^2}; \quad \dots; \quad y^{(r)}(x) = \frac{\Delta^r y_i}{h^r}. \quad (22a)$$

Лучшее приближение можно найти, учитывая, что в интерполяционном многочлене (17а) независимой переменной является q и поэтому

$$\frac{dy}{dx} = \frac{dy}{dq} \frac{dq}{dx} = \frac{1}{h} \frac{dy}{dq}.$$

Непосредственно дифференцируя многочлен (17а) по q , получают

$$\left. \begin{aligned} y'(x) &\approx \frac{1}{h} \left(\Delta y_0 + \frac{2q-1}{2} \Delta^2 y_0 + \frac{3q^2-6q+2}{6} \Delta^3 y_0 + \dots \right); \\ y''(x) &\approx \frac{1}{h^2} \left(\Delta^2 y_0 + (q-1) \Delta^3 y_0 + \frac{6q^2-18q+11}{12} \Delta^4 y_0 + \dots \right); \end{aligned} \right\} \dots \quad (22b)$$

Если аргумент производной близок к началу таблицы, то целесообразно принять начальным значение $x_0 = x_i$. Тогда $x_0 < x < x_1$ и формулы (22б) принимают более простой вид

$$\left. \begin{aligned} y'(x) &\approx \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \frac{\Delta^5 y_0}{5} - \dots \right); \\ y''(x) &\approx \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 - \frac{5}{6} \Delta^5 y_0 + \dots \right); \end{aligned} \right\} \dots \quad (22b)$$

В случаях, когда аргумент расположен ближе к концу таблицы, для вычисления производных используют многочлен (17б) для интерполяции назад.

Остаточную погрешность дифференцирования по формулам (22б) оценивают величиной

$$R \leq (-1)^m \frac{h^m}{m+1} y^{(m+1)}(\xi)$$

по наибольшему значению $(m+1)$ -й производной в интервале (x_0, x) . При достаточно малом шаге h модели остаточную погрешность приближенно оценивают по формуле

$$R \leq (-1)^m \frac{\Delta^{m+1} y_0}{h(m+1)}.$$

При кусочно-линейной аппроксимации табличной модели $y(x)$ значение интеграла в интервале между узлами равно площади трапеции:

$$\int_{x_i}^{x_{i+1}} y(x) dx \approx \frac{h}{2} (y_i + y_{i+1}). \quad (23a)$$

Следовательно, в интервале от x_0 до x_n значение интеграла, вычисленное по формуле трапеций,

$$I = \int_{x_0}^{x_n} y(x) dx \approx \frac{h}{2} (y_0 + 2y_1 + 2y_2 + \dots + 2y_{n-1} + y_n) \quad (23b)$$

будет тем точнее, чем меньше шаг h интегрирования. Остаточная погрешность в этом случае

$$R \leq -\frac{nh^3}{12} y''(\xi),$$

где ξ — значение аргумента, при котором вторая производная в интервале интегрирования максимальна.

Более точно интегрирование табличных моделей известных функций выполняется по параболической формуле (формуле Симпсона) для четного числа n отрезков интегрирования.

$$I = \int_{x_0}^{x_n} y(x) dx \approx \frac{h}{3} (y_0 + y_n + 4g_1 + 2g_2), \quad (24a)$$

где $g_1 = y_1 + y_3 + \dots + y_{n-1}$; $g_2 = y_2 + y_4 + \dots + y_{n-2}$.

Остаточная погрешность интегрирования по формуле (24a) определяется величиной

$$R \leq -\frac{h^5}{90} y^{(4)}(\xi),$$

где $y^{(4)}(\xi)$ — максимальное значение четвертой производной в интервале интегрирования. В связи с практической трудностью точного вычисления этой производной для уточнения результата интегрирования, повторно вычисляют интеграл по формуле (24a) с двойным шагом интегрирования $2h$. Для оценки точности верными полагают совпадающие знаки в значениях интегралов I_1 и I_2 , вычисленных с шагом h и $2h$. Интеграл уточняют по формуле $I = I_1 + (I_1 + I_2)/15$.

Формула Симпсона дает точные результаты при постоянстве вторых конечных разностей, когда табличная модель интерполируется многочленом второй или третьей степени. Функции,

аппроксимируемые многочленами до шестой степени включительно, достаточно точно интегрируются по формуле

$$I = \int_{x_0}^{x_6} y(x) dx \approx \frac{3h}{10}(y_0 + 5y_1 + y_2 + 6y_3 + y_4 + 5y_5 + y_6) \quad (246)$$

с остаточной погрешностью

$$R \leq -\frac{h^7}{1400}[10y^{(6)}(\xi) + 9h^2y^{(8)}(\xi)].$$

Функции, аппроксимируемые многочленами более высокой степени, интегрируют, разбивая интервал интегрирования на части, в которых применение формул (24) дает достаточно точные результаты.

Буквенные модели строят по полученной в результате эксперимента табличной модели различными способами. Наиболее употребительный из них заключается в аппроксимации (приближении) табличной модели первыми m членами сходящегося функционального ряда

$$\hat{f}(x) = \sum_{k=0}^m a_k f_k(x), \quad (25)$$

где a_k — коэффициенты; $f_k(x)$ — некоторые базисные функции.

Удобной мерой приближения аппроксимирующей функции $\hat{f}(x)$ к табличной модели в ее узлах является среднеквадратичное отклонение

$$\delta = \sum_{i=0}^n [y_i - \hat{f}(x_i)]^2, \quad (26)$$

величина которого тем меньше, чем лучше аппроксимация во всем интервале аргумента.

В качестве аппроксимирующей функции (25) часто используют степенной ряд (многочлен)

$$P(x) = \sum_{k=0}^m a_k x^k = a_0 + a_1 x + \dots + a_m x^m, \quad (27)$$

условия сходимости которого хорошо изучены. Такой многочлен наилучшим образом приближает табличную модель при равенстве нулю производных отклонения (26) по коэффициентам

$$-\frac{1}{2} \frac{\partial \delta}{\partial a_s} \equiv \sum_{i=0}^n \left[y_i - \sum_{k=0}^m a_k x_i^k \right] x_i^s = 0; \quad s = 0, 1, \dots, m.$$

Следовательно, коэффициенты a_k при подобном приближении определяются решением системы линейных уравнений

$$\sum_{i=0}^n x_i^s \sum_{k=0}^m a_k x_i^k = \sum_{i=0}^n y_i x_i^s, \quad s = 0, 1, \dots, m \quad (28a)$$

или, в матричной форме записи,

$$\begin{bmatrix} \sum x_i^0 & \dots & \sum x_i^m \\ \vdots & \ddots & \vdots \\ \sum x_i^m & \dots & \sum x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum y_i x_i^0 \\ \vdots \\ \sum y_i x_i^m \end{bmatrix},$$

где суммирование ведется по индексам $i = 0, 1, 2, \dots, n$ всех узлов табличной модели.

При линейной аппроксимации коэффициенты многочлена $a_0 + a_1 x$ определяются решением системы уравнений

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \end{bmatrix}, \quad (28b)$$

а коэффициенты квадратичного многочлена $a_0 + a_1 x + a_2 x^2$ — решением системы уравнений

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{bmatrix}. \quad (28b)$$

Среднеквадратичное приближение степенным многочленом улучшается при увеличении числа членов, но при этом возрастают затраты времени на вычисления. Поэтому обычно степень аппроксимирующего многочлена выбирают не больше порядка практически постоянных конечных разностей табличной модели.

Аппроксимирующие функции (в частности, степенные многочлены) можно построить методом неопределенных коэффициентов по условию совпадения их значений с табличными значениями в $m+1 \leq n+1$ узлах. Для этого выбирают систему базисных функций усечённого ряда (25), вычисляя его коэффициенты решением системы уравнений

$$y_j = \sum_{k=0}^m a_k f_k(x_j) \quad j = 0, 1, \dots, m \quad (29)$$

или, в матричной форме записи,

$$\begin{bmatrix} y_0 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} f_0(x_0) & \dots & f_m(x_0) \\ \dots & \dots & \dots \\ f_0(x_m) & \dots & f_m(x_m) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix}.$$

Полученная таким образом аппроксимирующая функция интерполирует табличную модель при $m+1$ узлах интерполяции, совпадающих с узлами модели, но отклонение такой функции от модели между узлами интерполяции будет большим, чем при вычислении коэффициентов многочлена из условия минимального среднеквадратичного отклонения.

Многие табличные модели хорошо приближаются эмпирическими формулами с двумя коэффициентами. Для определения вида такой функции вблизи границ интервала выбирают узлы x_a и x_b табличной модели с достоверными значениями $y(x_a) = y_a$ и $y(x_b) = y_b$ и вычисляют среднеарифметические $x_a = (x_a + x_b)/2$, $y_a = (y_a + y_b)/2$, среднегеометрические $x_r = \sqrt{x_a x_b}$, $y_r = \sqrt{y_a y_b}$ и среднегармонические $x_{rp} = 2x_a x_b / (x_a + x_b)$, $y_{rp} = 2y_a y_b / (y_a + y_b)$. Определив (при необходимости методами интерполирования) значения $y(x_a)$, $y(x_r)$ и $y(x_{rp})$, вычисляют модули их разностей со значениями y_a , y_r и y_{rp} и по наименьшему из этих модулей выбирают вид аппроксимирующей функции согласно табл. 7 [29].

Таблица 7

Минимальное значение	Аппроксимирующая функция	
$ y(x_a) - y_a $	Линейная	$y = ax + b$
$ y(x_a) - y_r $	Показательная	$y = ab^x$
$ y(x_r) - y_r $	Степенная	$y = ax^b$
$ y(x_r) - y_a $	Логарифмическая	$y = a \ln x + b$
$ y(x_{rp}) - y_a $	Гиперболическая	$y = a + b/x$
$ y(x_a) - y_{rp} $	Дробно-рациональная	$y = 1/(ax + b)$
$ y(x_{rp}) - y_{rp} $	Дробно-рациональная	$y = x/(ax + b)$

Коэффициенты a и b эмпирических формул вычисляют методом неопределенных коэффициентов по решению системы уравнений

$$y_a = f(x_a, a, b); \quad y_b = f(x_b, a, b)$$

или по минимальному среднеквадратичному отклонению (26) решением системы уравнений

$$\sum_{i=0}^n [y_i - f(x_i, a, b)] f'_a(x_i, a, b) = 0;$$

$$\sum_{i=0}^n [y_i - f(x_i, a, b)] f'_b(x_i, a, b) = 0.$$

Во многих инженерных приложениях для аппроксимации табличных моделей с интервалом $T = x_n - x_0$ функциональным рядом (25) используют усеченный тригонометрический ряд (ряд Фурье)

$$f(x) = \sum_{k=-m}^m C_k e^{-ik\omega_0 x} = C_0 + 2 \sum_{k=1}^m |C_k| \cos(k\omega_0 x + \varphi_k) = \\ = \frac{a_0}{2} + \sum_{k=1}^m (a_k \cos k\omega_0 x + b_k \sin k\omega_0 x), \quad (30)$$

где $\omega_0 = 2\pi/T$; $\varphi_k = -\arctg b_k/a_k$; $C_k = \frac{1}{T} \int_0^T y(x) e^{-ik\omega_0 x} dx =$

$$= \sqrt{a_k^2 + b_k^2} e^{i\varphi_k}; \quad a_k = \frac{2}{T} \int_0^T y(x) \cos k\omega_0 x dx; \quad b_k = \frac{2}{T} \int_0^T y(x) \times$$

$$\times \sin k\omega_0 x dx.$$

При ограничении ряда (30) m членами и числе $n \geq 2m$ точек табличной модели такой усеченный ряд дает наилучшее приближение в смысле среднеквадратичного отклонения (26), если его коэффициенты вычислены по формулам

$$a_k = \frac{2}{n} \sum_{i=0}^{n-1} y_i \cos \frac{2\pi ki}{n}; \quad b_k = \frac{2}{n} \sum_{i=0}^{n-1} y_i \sin \frac{2\pi ki}{n} \quad (31a)$$

для n точек табличной модели с аргументами $x_i = iT/n$ в равнотстоящих узлах с индексами $i = 0, 1, \dots, n-1$.

Коэффициенты ряда Фурье (гармонический анализ) по заданной табличной модели с шагом $h = T/n$ определяют также по формулам дискретного преобразования Фурье (ДПФ)

$$C_k = h \sum_{i=0}^{n-1} y(ih) e^{-2\pi ik/n}; \quad k = 0, 1, \dots, n-1. \quad (31b)$$

Для ускорения вычислений по этим формулам используют алгоритмы быстрого преобразования Фурье (БПФ), но они мало пригодны для ручных вычислений.

4. ПРЕОБРАЗОВАНИЯ БУКВЕННЫХ МОДЕЛЕЙ

Символические (буквенные) модели физических зависимостей удобны для компактной записи аппроксимирующей функции в непрерывном интервале аргумента. Вычисление таких функций для заданного значения аргумента в конечном итоге сводится к последовательности арифметических операций. Для этого буквенные модели, содержащие символы элементарных и специальных функций и функционалов, предварительно преобразуют в формулы, со-

держащие в общем случае только символы операндов и арифметических операций (возвведение в степень в этом случае рассматривается как многократное умножение)*.

Расчетные формулы непосредственно получаются при аппроксимации табличных моделей степенными многочленами или другими функциями, вычисление которых сводится к арифметическим операциям. Расчетную формулу для заданной сложной функции можно построить по ее предварительно составленной табличной модели, но в этом случае необходимо вычислять значения функции, заданной буквенной моделью, в узлах моделирующей таблицы.

Среди методов построения расчетных формул, не требующих построения табличной модели функции, наиболее употребительны представления степенными многочленами по разложениям в ряд Тейлора, дробно-рациональные приближения, представления цепными дробями, а также приближенные методы вычисления функций, связанные с использованием итерационных формул.

Аналитическая (имеющая все производные в интервале изменения аргумента) функция, заданная буквенной моделью, представима степенным рядом в виде разложения функции в окрестности значения x_0 аргумента в ряд Тейлора

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k = f(x_0) + \frac{f'(x_0)}{1!} (x - x_0) + \dots \quad (32)$$

с остаточной погрешностью при m членах ряда

$$R = \frac{f^{(m+1)}(x_0 + \theta x - \theta x_0)}{(m+1)!} (x - x_0)^{m+1}; \quad 0 < \theta < 1.$$

Преимущества представления аналитических функций рядом Тейлора связаны с возможностью вычисления очередного члена ряда непосредственно по заданной функции без вычисления предыдущих коэффициентов. Недостатки такого разложения заключаются в медленной сходимости ряда для некоторых функций, что увеличивает время вычислений и операционную погрешность округления промежуточных результатов. Кроме того, при увеличении аргумента возрастают остаточные погрешности, и для их уменьшения аргумент приходится предварительно преобразовывать в более узкую область.

Пример 6. Представить функции \sqrt{x} и $\sin x$ степенными рядами в интервале аргумента от 0 до ∞ .

Выберем целое число y_0 , квадрат которого будет ближайшим к числу x . Тогда

$$\sqrt{x} = \sqrt{y_0^2 - y_0^2 + x} = y_0 \sqrt{1 + (x - y_0^2)/y_0^2} = y_0 \sqrt{1 + z} = y_0 f(z),$$

где $z = (x - y_0^2)/y_0^2$, причем $|z| < 1$, а $f(z) = 1$ при $z = 0$.

* При использовании таблиц элементарных функций или микрокалькуляторов, автоматически вычисляющих их значения, расчетные формулы могут содержать и символы этих функций.

Производные $f(z)$ при $z = 0$:

$$f^1(0) = 1/2; \quad f^{11}(0) = -1/4; \quad f^{111}(0) = 3/8; \quad f^{IV}(0) = -15/16; \dots;$$

$$f^{(n)}(0) = (-1)^{n-1} 1 \cdot 3 \cdot 5 \cdots (2n-3)/2.$$

В соответствии с формулой (32) получаем искомое разложение

$$Vx = y_0 \left(1 + \frac{z}{1!2} - \frac{z^2}{2!4} + \frac{3z^3}{3!8} - \frac{3 \cdot 5z^4}{4!16} \cdots \right).$$

Например для $x = 10$; $y_0 = 3$; $z = 1/9$ вычисляем

$$\sqrt{10} = 3 \left(1 + \frac{1}{18} - \frac{1}{648} + \frac{1}{11664} - \frac{5}{839808} \right) \approx 3,16228.$$

Проверка дает $3,16228^2 = 10,000001$.

Разложение функции $\sin x$ в ряд Тейлора вблизи точки x_0 имеет вид

$$\sin x = \sin x_0 + \frac{\cos x_0}{1!} (x - x_0) - \frac{\sin x_0}{2!} (x - x_0)^2 - \dots$$

Для ускорения сходимости вычислений разность $x - x_0$ обычно ограничивают значением $x - x_0 \leq \pi/4$. В этом случае в зависимости от выбранного значения x_0 искомое разложение

$$\sin x = x - x_0 - \frac{(x - x_0)^3}{3!} + \frac{(x - x_0)^5}{5!} - \dots \text{ при } x_0 = k\pi;$$

$$\sin x = \pm \left[1 - \frac{(x - x_0)^2}{2!} + \frac{(x - x_0)^4}{4!} - \dots \right] \text{ при } x_0 = k\pi \pm \frac{\pi}{2};$$

$$\sin x = \frac{\pm i}{\sqrt{2}} \left[1 \pm (x - x_0) + \frac{(x - x_0)^2}{2!} \pm \frac{(x - x_0)^3}{3!} + \dots \right] \text{ при } x_0 = k\pi \pm \frac{\pi}{2} \pm \frac{\pi}{4},$$

где k — целое число.

Значение степенного многочлена (27) для заданного значения аргумента удобно находить цепными вычислениями по формуле

$$P(x) = (\dots (a_m x + a_{m-1}) x + \dots + a_1) x + a_0, \quad (33)$$

соответствующей схеме Горнера для заданного значения $x = \xi$:

$$\begin{array}{r} + a_m & a_{m-1} & a_{m-2} & \dots & a_0 \\ \hline a_m \xi & b_{m-1} \xi & b_{m-2} \xi & \dots & b_1 \xi \\ \hline a_m & b_{m-1} & b_{m-2} & \dots & b_0 \end{array} = P(\xi).$$

При ручных вычислениях в первую строку схемы записывают коэффициенты многочлена в порядке убывания степеней, переносят a_m в третью строку, остальные элементы которой вычисляют умножением предыдущего элемента b_i на ξ и сложением произведения с a_i . Подобные вычисления повторяют до нахождения всех коэффициентов b_i , получая значение многочлена $P(\xi) = b_0$.

В процессе вычислений по схеме Горнера определяются коэффициенты b_m, \dots, b_1 многочлена $P_1(x)$, равного частному от деления исходного многочлена $P(x)$ на двучлен $(x - \xi)$ согласно тождеству $P(x) = (x - \xi) P_1(x)$. Таким образом, коэффициенты b_i в схеме Горнера равны содержимому соответствующих пар круглых скобок в формуле (33).

Если деление по схеме Горнера повторить для многочленов $P_1(x), \dots, P_m(x)$, то после каждого деления будут получены значения $b_{0k} = P_k(x)$ коэффициентов многочлена $P(y + \xi)$ той же степени, полученного заменой $x = y + \xi$ в исходном многочлене. На этом основана обобщенная схема Горнера, в которой вычисления по основной схеме повторяются до получения всех m коэффициентов b_{0k} многочлена $P(y + \xi)$.

Пример 7. Заменить переменную $x = y + 1$ в многочлене $P(x) = 2x^5 + 3x^4 - 4x^3 + 2x^2 - 2x + 4$.

Согласно обобщенной схеме Горнера последовательно вычисляем

$$\begin{array}{r}
 \begin{array}{ccccccc}
 + & 2 & 3 & -4 & 2 & -2 & 4 & 1 \\
 + & & 2 & 5 & 1 & 3 & 1 & 5 \\
 \hline
 + & 2 & 5 & 1 & 3 & 1 & 5 \\
 + & 2 & 7 & 8 & 11 & & \\
 \hline
 + & 2 & 7 & 8 & 11 & 12 \\
 + & 2 & 9 & 17 & 28 \\
 \hline
 + & 2 & 11 & 28 \\
 \hline
 & 13 \\
 \hline
 & 2
 \end{array}
 \end{array}$$

Следовательно, $P(y + 1) = 2y^5 + 13y^4 + 28y^3 + 28y^2 + 12y + 5$.

При рассмотренных представлениях функций вычисления сводятся к суммированию сходящихся числовых рядов с суммой членов

$$S = \sum_{k=1}^{\infty} u_k = u_1 + u_2 + \dots + u_m + R_m, \quad (34)$$

где остаточная погрешность суммирования ряда положительных монотонно убывающих значений функции ограничена пределами

$$\int_{m+1}^{\infty} f(x) dx \leq R_m \leq \int_m^{\infty} f(x) dx,$$

а для знакопеременного ряда с монотонно убывающими по модулям членами $|R_m| \leq |u_{m+1}|$, причем знак остатка совпадает со знаком члена u_{m+1} .

Предельная погрешность суммы m членов ряда (34) $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$ складывается из остаточной погрешности ε_1 , операционной погрешности округления частных сумм ε_2 и погрешности ε_3 округления результата. При заданной предельной погрешности ε обычно принимают $\varepsilon_1 = \varepsilon_2 = \varepsilon/4$, $\varepsilon_3 = \varepsilon/2$ и по полученному значению ε_1 определяют требуемое число членов ряда, а по ε_2 и ε_3 — требуемые точности представления частичных сумм и окончательного результата. Для заданной точности числа до k верных знаков после запятой при округлении k -ю цифру оставляют без изменений,

если следующая не больше 3, и увеличивают на единицу, если следующая цифра не менее 6. Если она равна 4 или 5, то точность представления числа увеличивают на один разряд.

Пример 8. Найти сумму ряда $S = \sum_{k=1}^m 1/9k^3$ с точностью 0,001.

Примем предельную погрешность остатка $\varepsilon_1 \leq 0,25 \cdot 10^{-3}$. Тогда

$$R_m \leq \int_m^\infty \frac{dx}{9x^3} = \frac{1}{18m^2} \leq \varepsilon_1,$$

откуда $m \geq 14,91 \approx 15$. При $\varepsilon_2 = 0,25 \cdot 10^{-3}$ допустимая предельная погрешность частных сумм $\varepsilon_1 = \varepsilon_2/m \leq 0,25 \cdot 10^{-3}/15 = 1,67 \cdot 10^{-5}$ и, следовательно, частные суммы необходимо вычислять с точностью до пяти верных знаков после запятой.

Разбивая слагаемые на три группы для уменьшения погрешности, находим

$$\begin{aligned} 0,11111 + 0,01389 + 0,00412 + 0,00174 + 0,00089 &= 0,13174; \\ 0,00051 + 0,00033 + 0,00022 + 0,00015 + 0,00011 &= 0,00132; \\ 0,00008 + 0,00006 + 0,00005 + 0,00004 + 0,00003 &= 0,00026. \end{aligned}$$

Складывая частные суммы и округляя с требуемой точностью, получаем $S = 0,133$.

Недостаток представления степенными рядами, как подтверждается и рассмотренным примером, заключается в медленной сходимости членов ряда.

Для представления степенного многочлена вида (27) дробно-рациональным приближением его приравнивают дробно-рациональной функции

$$\frac{b_0 + b_1x + \dots + b_rx^r}{1 + c_1x + \dots + c_sx^s} = a_0 + a_1x + \dots + a_mx^m \quad (35)$$

со степенями числителя и знаменателя $r = m/2$, $s = m/2$ при четном m и $r = (m+1)/2$, $s = (m-1)/2$ при нечетном m . Приведя равенство (35) к общему знаменателю, приравнивают члены с одинаковыми степенями x в правой и левой части равенства, получая систему линейных уравнений

$$\begin{aligned} b_0 &= a_0; \\ b_1 &= a_1 + a_0c_1; \\ &\dots \\ b_r &= a_r + a_{r-1}c_1 + \dots + a_0c_s; \\ 0 &= a_{r+1} + a_rc_1 + \dots + a_1c_s; \\ 0 &= a_{r+2} + a_{r+1}c_1 + \dots + a_2c_s; \\ &\dots \\ 0 &= a_m + a_{m-1}c_1 + \dots + a_sc_s. \end{aligned}$$

Решив последние s уравнений относительно коэффициентов c_i , подставляют их значения в остальные уравнения и вычисляют коэффициенты b_i .

Представление степенного ряда дробно-рациональной функцией не уменьшает числа операций при вычислении функции по заданному значению аргумента по сравнению с вычислением степенного многочлена по схеме Горнера. Однако при преобразовании дробно-рациональной функции в цепную дробь при вычислении последней может потребоваться меньшее число операций.

Цепной дробью называют разложение вида

$$f = \alpha_0 + \frac{\beta_1}{\alpha_1 + \frac{\beta_2}{\alpha_2 + \frac{\beta_3}{\alpha_3 + \dots}}} = \left[\alpha_0; \frac{\beta_1}{\alpha_1}; \frac{\beta_2}{\alpha_2}; \dots \right], \quad (36)$$

где справа приведена условная запись цепной дроби общего вида. Числа α_k , β_k называют элементами цепной дроби, а каждую k -ю пару коэффициентов β_k и α_k — звеном цепной дроби.

Цепную дробь можно преобразовать к другому виду, в связи с чемодну и ту же функцию можно представить различными цепными дробями, отличающимися значениями элементов и числом звеньев. В частности, цепную дробь общего вида (36) с конечным или бесконечным числом звеньев можно преобразовать в обыкновенную цепную дробь с элементами $\beta_k = 1$.

Дробно-рациональную функцию, эквивалентную усеченному степенному ряду согласно соотношению (35), можно представить цепной дробью различными способами. Один из них основан на приравнивании дробно-рациональной функции и цепной дроби с неизвестными элементами согласно формуле

$$f(x) = \frac{a_{10} + a_{11}x + a_{12}x^2 + \dots}{a_{00} + a_{01}x + a_{02}x^2 + \dots} = \left[0, \frac{a_{10}}{a_{00}}, \frac{a_{20}x}{a_{10}}, \frac{a_{30}x}{a_{20}}, \dots \right]. \quad (37a)$$

Неизвестные элементы цепной дроби a_{i0} в этой формуле являются элементами матрицы

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \dots \\ a_{10} & a_{11} & a_{12} & \dots \\ a_{20} & a_{21} & a_{22} & \dots \\ a_{30} & a_{31} & a_{32} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix},$$

первые две строки которой образуются коэффициентами числителя и знаменателя заданной функции, а элементы остальных строк вычисляют по формуле

$$a_{ij} = \begin{vmatrix} a_{i-1,0} & a_{i-1,1} & \dots & a_{i-1,j+1} \\ a_{i-2,0} & a_{i-2,1} & \dots & a_{i-2,j+1} \end{vmatrix} = a_{i-1,0} a_{i-2,1} - a_{i-2,0} a_{i-1,1}. \quad (37b)$$

Если при вычислении этих элементов окажется $a_{k0} = 0$, то $(k+1)$ -я строка образуется сдвигом влево на один столбец элементов k -й строки.

Заданную дробно-рациональную функцию $f(x)$ можно также разложить в обыкновенную цепную дробь почленным делением (на-

чиная с младших или старших степеней) числителя на знаменатель исходной функции или знаменатель на числитель инвертированной функции $f(x) = 1/(1/f(x))$ до получения первого члена частного, после чего предыдущий делитель делят до получения первого члена частного на остаток предыдущего деления. Подобную процедуру повторяют до разложения заданной функции. Каждое частное от деления равно очередному элементу a_i обыкновенной цепной дроби. Порядок членов делимого и делителя можно изменять в процессе разложения, но это приводит к более громоздким цепным дробям.

Пример 9. Представить цепной дробью функцию

$$f(x) = (x+2)/(x^2 + 2x + 1).$$

В соответствии с формулами (37) составляем матрицу

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ 3 & 2 & 0 \\ -1 & 0 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

и цепную дробь

$$f(x) = \left[0, \frac{2}{1}, \frac{3x}{2}, \frac{-x}{3}, \frac{2x}{1} \right] = \frac{2}{1 + \frac{3x}{2 - \frac{x}{3 + 2x}}}.$$

При разложениидробно-рациональной функции почленным делением удобно использовать приводимую ниже схему, в которой делимое отделено обратными скобками от делителя и частного. При разложении по этой схеме заданной функции, начиная со старших степеней, получаем

$$\begin{aligned} & x^2 + 2x + 1 \Big| x + 2 \quad \left(\frac{1}{x} \right) \\ & \quad - x + 2 + \frac{1}{x} \\ & \quad \overline{\quad \quad \quad - \frac{1}{x}} \Big| x^2 + 2x + 1 \quad \left(-x^3 \right) \\ & \quad \quad \quad \overline{-x^2} \\ & \quad \quad \quad - 2x + 1 \quad \left(-\frac{1}{2x^2} \right) \\ & \quad \quad \quad \overline{-x} - \frac{1}{2x^2} \\ & \quad \quad \quad \overline{\quad \quad \quad \frac{1}{2x^2}} \Big| 2x + 1 \quad (4x^3) \\ & \quad \quad \quad - 2x \quad \left(\frac{1}{2x^2} \right) \\ & \quad \quad \quad \overline{\quad \quad \quad 1) \frac{1}{2x^2}} \Big| \frac{1}{2x^2} \\ & \quad \quad \quad \overline{\quad \quad \quad \frac{1}{2x^2}} \\ & \quad \quad \quad \overline{\quad \quad \quad 0}. \end{aligned}$$

Следовательно,

$$f(x) = \frac{1}{x} + \frac{1}{-x^3 + \frac{1}{-\frac{1}{2x^2} + \frac{1}{4x^3 + 2x^2}}} = \left[\frac{1}{x}, \frac{1}{-x^3}, \frac{1}{-1/2x^2}, \frac{1}{4x^3}, \frac{1}{1/2x^2} \right].$$

При разложении $f(x)$, начиная с младших степеней, аналогично получаем

$$f(x) = 2 + \frac{1}{-\frac{1}{3x} + \frac{1}{-\frac{9}{5} + \frac{1}{\frac{16}{3x} + 4}}} = \left[2, \frac{1}{-1/3x}, \frac{1}{-9/5}, \frac{1}{16/3x^2}, \frac{1}{1/4} \right].$$

При разложении знаменателя инвертированной функции $f(x) = 1/(x^2 + 2x + 1)/(x + 2)$, начиная со старших степеней, получаем

$$\begin{array}{r} x+2 \quad \underline{-x^2+2x+1}(x) \\ \underline{x^2+2x} \end{array} \begin{array}{l} 1) \quad x+2(x) \\ \underline{x} \\ 2) \quad -1\left(\frac{1}{2}\right) \\ \underline{-1} \\ 0 \end{array}$$

и, следовательно,

$$f(x) = \frac{1}{x+\frac{1}{x+2}} = \left[0, \frac{1}{x}, \frac{1}{x}, \frac{1}{1/2} \right].$$

При разложении инвертированной функции, начиная с младших степеней, аналогично получаем

$$\begin{array}{r} 1 \quad \underline{\frac{1}{2}+\frac{1}{4}} \\ \underline{3x+\frac{1}{-\frac{9}{2}+\frac{1}{3x}}} \end{array} = \left[0, \frac{1}{1/2}, \frac{1}{4/3x}, \frac{1}{-9/2}, \frac{1}{-1/3x} \right].$$

Вычисление функции, представленной разложением в цепную дробь с большим или, теоретически, бесконечным числом звеньев, основано на замене цепной дроби k -й подходящей дробью. Подходящей k -й дробью называют обыкновенную дробь, числитель и знаменатель которой вычисляют по рекуррентной формуле

$$\frac{a_k}{b_k} = \frac{a_k a_{k-1} + b_k a_{k-2}}{a_k b_{k-1} + b_k b_{k-2}}, \quad (38)$$

где в качестве начальных приближений для правой части принимают $a_{-1} = 1; b_{-1} = 0; a_0 = \alpha_0$ и $b_0 = 1$. Цепная дробь, ограниченная m звеньями, приближенно равна m -й подходящей дроби.

Преимущества представления функций цепной дробью заключаются в относительно широкой области сходимости последней и простоте вычисления k -х подходящих дробей. Недостатки разложения в ряд Тейлора и последующих представлений степенного

ряда дробно-рациональной функцией или цепной дробью связаны с медленной сходимостью такого ряда для некоторых функций, что приводит к затратам времени на вычисления и к возрастанию операционной погрешности при увеличении числа членов ряда при заданной остаточной погрешности. Кроме того, остаточная погрешность ряда Тейлора и последующих его эквивалентных представлений, равная нулю при $x = x_0$, возрастает по мере увеличения разности $x - x_0$.

Более равномерно распределена по интервалу аргумента остаточная погрешность аппроксимации степенным многочленом (и других его представлений) при вычислении коэффициентов такого многочлена методом неопределенных коэффициентов и, особенно, по критерию минимального среднеквадратичного отклонения [выражение (26)]. Однако и в этом случае такая погрешность распределена неравномерно по интервалу аргумента и обычно возрастает на его краях.

Наиболее эффективна равноволновая (чебышевская) аппроксимация, при которой минимизируются наибольшие отклонения от аппроксимируемой функции в интервале аргумента. Равноволновая аппроксимация связана с использованием многочленов Чебышева n -го порядка

$$T_n(x) = \cos(n \arccos x), \quad (39a)$$

вычисляемых по рекуррентной формуле

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad (39b)$$

где $T_0(x) = 1$; $T_1(x) = x$.

Построение равноволновой аппроксимирующей функции достаточно сложно, но во многих случаях сравнительно простые способы использования многочленов Чебышева позволяют повысить точность аппроксимации или уменьшить число элементов аппроксимирующего выражения при заданной точности.

Для уменьшения погрешностей степенных многочленов (27), полученных разложением в ряд Тейлора, используют метод экономизации. Он заключается в выражении высшей степени x в m -м члене через многочлены Чебышева согласно формулам:

$$\begin{aligned} x &= T_1; \\ x^2 &= (1 + T_2)/2; \\ x^3 &= (3x + T_3)/4; \\ x^4 &= (8x^2 - 1 + T_4)/8; \\ x^5 &= (20x^3 - 5x + T_5)/16; \\ x^6 &= (48x^4 - 18x^2 + 1 + T_6)/32; \\ x^7 &= (112x^5 - 56x^3 + 7x + T_7)/64; \\ x^8 &= (256x^6 - 160x^4 + 32x^2 - 1 + T_8)/128; \\ x^9 &= (576x^7 - 432x^5 + 120x^3 - 9x + T_9)/256; \\ x^{10} &= (1280x^8 - 1120x^6 + 400x^4 - 50x^2 + 1 + T_{10})/512; \\ x^{11} &= (2816x^9 - 2816x^7 + 1232x^5 - 220x^3 + 11x + T_{11})/1024, \end{aligned} \quad (40)$$

после чего отбрасывают член, содержащий T_k . Отбрасывание вносит погрешность $\varepsilon < a_m 2^{-(k-1)}$, но она частично компенсируется уменьшением операционной погрешности и более равномерным распределением остаточной погрешности по интервалу аргумента. Полученный многочлен с $(m-1)$ членами низшей степени даст лучшее приближение аппроксимируемой функции, чем исходный ряд Тейлора при усечении до $m-1$ членов. Повторив экономизацию для нескольких старших степеней усеченного ряда Тейлора, получают аппроксимирующий многочлен с относительно небольшим числом членов и более равномерным распределением остаточной погрешности.

Для вычисления некоторых функций целесообразно использовать итерационные алгоритмы. Если требуется вычислить значение непрерывно дифференцируемой в интервале аргумента функции $y = f(x)$, то для заданного значения аргумента ее можно представить в неявном виде уравнением $\varphi(x, y) = 0$. При приближенном значении функции y_k справедливо равенство $\varphi(x, y_k) = (y_k - y) \varphi'_y(x, \bar{y}_k)$, где \bar{y}_k — промежуточное между y и y_k значение функции. Отсюда следует $y = y_k - \varphi(x, y_k)/\varphi'_y(x, \bar{y}_k)$. Полагая $y \approx \bar{y}_k$, приходим к итеративному процессу, описываемому формулой Ньютона

$$y_{k+1} = y_k - \frac{\varphi(x, y_k)}{\varphi'_y(x, y_k)}; \quad k = 0, 1, 2, \dots \quad (41)$$

Сходимость такого бесконечного процесса к искомому значению функции обеспечивается в том случае, когда производные $\varphi'_y(x, y)$ и $\varphi''_y(x, y)$ сохраняют постоянным знак в интервале, содержащем точное значение y .

Для примера составим формулу, описывающую алгоритм вычисления $y = \sqrt{x}$. Приняв $\varphi(x, y) \equiv y^2 - x = 0$, получим $\varphi'_y(x, y) = 2y$ и в соответствии с формулой (41)

$$y_{k+1} = \frac{1}{2} \left(y_k + \frac{x}{y_k} \right); \quad k = 0, 1, 2, \dots \quad (42a)$$

При вычислениях по этой формуле (называемой формулой Герона) число верных цифр результата примерно удваивается на каждой итерации и процесс быстро сходится.

Аналогично составляют формулы итерационного процесса для вычисления обратной функции $y = 1/x$ без деления

$$y_{k+1} = y_k (2 - xy_k); \quad k = 0, 1, 2, \dots \quad (42b)$$

и вычисления функции $y = 1/\sqrt{x}$

$$y_{k+1} = y_k (3 - xy_k^2)/2; \quad k = 0, 1, 2, \dots \quad (42b)$$

Построение итеративного процесса для вычисления кубического корня $y = \sqrt[3]{x}$ приводит к формуле

$$y_{k+1} = \frac{1}{3} \left(2y_k + \frac{x}{y_k^2} \right); \quad k = 0, 1, 2, \dots, \quad (42c)$$

дающей для y_{k+1} число верных цифр $2p$ или $2p - 1$, если y_k имеет p верных цифр.

Эффективность итеративных процессов неодинакова для различных функций с различными значениями аргумента и параметров. Например, для степенной функции x^a способ вычисления зависит от значения аргумента и показателя степени. Для отрицательных целых чисел и правильных дробей $a = 1/b$, где $b = 2, 3, 4, \dots$, лучший способ, как правило, заключается в использовании итерационного процесса. Однако для иррациональностей с положительной неправильной дробью в показателе лучшим может оказаться представление цепной дробью

$$(1+x)^a = \left[1, \frac{ax}{1}, \frac{(1-a)x}{2}, \frac{(1+a)x}{3}, \dots, \frac{(k-a)x}{2k}, \frac{(k+a)x}{2k+1}, \dots \right], \quad (43a)$$

сходящейся на всей плоскости комплексного аргумента x за исключением разреза по вещественной оси от $-\infty$ до -1 .

При $|x| < 1$ целесообразно использовать биномиальное разложение

$$\begin{aligned} (1+x)^a = 1 + ax + \frac{a(a-1)}{2!}x^2 + \dots + \\ + \frac{a(a-1)\dots(a-k+1)}{k!}x^k + \dots, \end{aligned} \quad (43b)$$

где a может быть дробным и отрицательным числом.

Показательную функцию общего вида обычно вычисляют по формуле

$$a^x = 1 + \ln a \cdot x + \frac{\ln^2 a}{2!}x^2 + \dots + \frac{\ln^k a}{k!}x^k + \dots, \quad (44)$$

где $\ln a$ определяют по таблицам логарифмов или вычисляют по приводимым ниже формулам.

Экспоненциальную функцию при $|x| < 1$ вычисляют при помощи ряда

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2} + \dots + \frac{x^k}{k!} + \dots + R_m \quad (45a)$$

с остаточной погрешностью

$$R_m = \frac{e^{\theta} x^{m+1}}{(m+1)!}; \quad 0 < \theta < 1.$$

При $|x| > 1$ аргумент экспоненциальной функции представляют суммой $x = x_1 + x_2$ целой и дробной частей, а функцию — произведением $e^x = e^{x_1} \cdot e^{x_2}$. Первый множитель вычисляют умножением x_1 раз числа $e = 2,718\ 281\ 824\ 590\ 45\dots$, если $x > 0$, или $1/e = 0,367\ 879\ 441\ 171\ 44\dots$, если $x < 0$. Второй множитель вычисляют с помощью быстро сходящегося для малых значений x_2 ряда (45a).

Экспоненциальную функцию вычисляют также по цепным дробям

$$e^x = \left[1; \frac{x}{1}, -\frac{x}{2}, \frac{x}{3}, \dots, -\frac{x}{2}, \frac{x}{2k+1}, \dots \right]; \quad (45a)$$

$$e^x = \left[1; \frac{2x}{2-x}, \frac{x^2}{6}, \frac{x^2}{10}, \dots, \frac{x^2}{2(2k+1)}, \dots \right], \quad (45b)$$

сходящимся на всей плоскости комплексного переменного x .

Логарифмическую функцию можно вычислить по разложению в цепную дробь

$$\ln(1+x) = \left[0; \frac{x}{1}, \frac{x}{2}, \frac{x}{3}, \dots, \frac{kx}{2}, \frac{kx}{2k+1}, \dots \right], \quad (46a)$$

сходящуюся на всей плоскости комплексного аргумента за исключением разреза вдоль вещественной оси от $-\infty$ до -1 .

При $|x| \leq 1$ логарифмическую функцию представляют рядом

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5} - \dots, \quad (46b)$$

но этот ряд сходится медленно и пригоден лишь для малых значений аргумента. Поэтому положительные значения аргумента представляют как $x = 2^a z$, где a — целое число, а $0,5 < z < 1$, и используют разложение

$$\ln x = \ln 2^a z = a \ln 2 - 2 \left(w + \frac{w^3}{3} + \dots + \frac{w^{2k-1}}{2k-1} + \dots \right), \quad (46b)$$

где $\ln 2 = 0,693\ 147\ 18\dots$, а $w = (1-z)/(1+z)$. Процесс суммирования ряда в скобках формулы (46b) ограничивают членом $u_m < 4\varepsilon$, где ε — заданная допустимая остаточная погрешность вычисления $\ln x$.

Десятичные логарифмы вычисляют по формуле

$$\lg x = M \ln x, \quad (47)$$

где $M = 0,434\ 294\ 481\ 903\dots$, а $\ln x$ выражается формулой (46b).

При вычислениях тригонометрических функций большие значения аргумента с помощью формул приведения (или приемов, подобных рассмотренному в примере 6) обычно заключают в интервал $0 \leq x \leq \pi/2$. Для вычисления функций $\sin x$ и $\cos z = \cos(\pi/2 - x)$ в интервале $0 \leq x \leq \pi/4$ часто используют степенной ряд

$$\cos z = \sin x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots, \quad (48a)$$

а в интервале $\pi/4 \leq x \leq \pi/2$ — степенной ряд

$$\sin x = \cos z = \sum_{k=0}^{\infty} (-1)^k \frac{z^{2k}}{(2k)!} = 1 - \frac{z^2}{2!} + \frac{z^4}{4!} - \dots \quad (48b)$$

или представления этих рядов цепными дробями. Остаточная погрешность этих рядов не превышает по модулю первого члена остатка, но возрастает при увеличении аргумента. Экономизацией рядов можно уменьшить число членов, обеспечивающих заданную точность вычислений.

Значение тангенса при $|x| \leq \pi/2$ часто вычисляют при помощи ряда

$$\operatorname{tg} x = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \frac{62x^9}{2835} + \dots, \quad (48\text{в})$$

но более удобно представление цепной дробью

$$\operatorname{tg} x = \left[0; \frac{x}{1}, \frac{-x^2}{3}, \frac{-x^2}{5}, \dots, \frac{-x^2}{2k+1}, \dots \right], \quad (48\text{г})$$

сходящейся на всей комплексной плоскости аргумента за исключением тех ее точек, где $\operatorname{tg} x$ обращается в бесконечность.

Гиперболические синус и косинус обычно вычисляют при помощи рядов

$$\operatorname{sh} x = \frac{e^x - e^{-x}}{2} = \sum_{k=0}^{\infty} \frac{x^{2k+1}}{(2k+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots \quad (49\text{а})$$

с остаточной погрешностью $R_m \leq 2u_m/3$ и

$$\operatorname{ch} x = \frac{e^x + e^{-x}}{2} = \sum_{k=0}^{\infty} \frac{x^{2k}}{(2k)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots \quad (49\text{б})$$

с остаточной погрешностью $R_m \leq 2u_m/3$. Эти ряды только знаками отличаются от рядов (48а) и (48б) соответственно.

Гиперболический тангенс вычисляют по цепной дроби

$$\operatorname{th} x = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \left[0; \frac{x}{1}, \frac{x^2}{3}, \frac{x^2}{5}, \dots, \frac{x^2}{2k-1}, \dots \right], \quad (49\text{в})$$

но при больших значениях $x > 0$ целесообразно использовать формулу

$$\operatorname{th} x = 1 - \frac{2}{e^{2x} + 1}, \quad (49\text{г})$$

а при малых значениях $|x| \leq \pi/2$ — степенной ряд

$$\operatorname{th} x = x - \frac{x^3}{3} + \frac{2x^5}{15} - \frac{17x^7}{315} + \frac{62x^9}{2835} - \dots \quad (49\text{д})$$

Обратные тригонометрические и гиперболические функции вычисляют при помощи сходящихся в интервале $-1 \leq x \leq 1$ рядов

$$\arcsin x = x + \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots; \quad (50a)$$

$$\operatorname{arctg} x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \dots; \quad (50b)$$

$$\operatorname{arsh} x = x - \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} - \frac{1 \cdot 3 \cdot 5 x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots; \quad (50c)$$

$$\operatorname{arth} x = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \frac{x^9}{9} + \dots \quad (50d)$$

При вычислении обратных тригонометрических функций часто используют связывающие их соотношения:

$$\begin{aligned} \arcsin x &= \operatorname{arctg} (x / \sqrt{1 - x^2}); \quad \arccos x = \pi/2 - \arcsin x; \\ \operatorname{arctg} x &= \pi/2 - \operatorname{arcctg} x = \pi/2 - \operatorname{arctg} 1/x. \end{aligned} \quad (51)$$

В частности, при $x > 1$ используют ряд

$$\operatorname{arctg} x = \frac{\pi}{2} - \operatorname{arctg} 1/x = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \frac{1}{7x^7} - \dots. \quad (52a)$$

Разложения в цепные дроби

$$\operatorname{arctg} x = \left[0; \frac{x}{1}, \frac{x^2}{3}, \frac{4x^2}{5}, \dots, \frac{k^2 x^2}{2k+1}, \dots \right]; \quad (52b)$$

$$\operatorname{arctg} x = \left[x; \frac{-x}{3}, \frac{9x^2}{5}, \frac{4x^2}{7}, \dots, \frac{(2k+1)^2 x^2}{4k+1}, \frac{(2k)^2 x^2}{4k+3}, \dots \right] \quad (52c)$$

относительно быстро сходятся на всей плоскости аргумента за исключением разрезов по мнимой оси от $-j\infty$ до $-j1$ и от $j\infty$ до $j1$.

Функции Бесселя порядка n вычисляют по разложению в степенной ряд

$$J_n(x) = \frac{1}{n!} \left(\frac{x}{2}\right)^n \left[1 - \frac{\left(\frac{x}{2}\right)^2}{1!(n+1)} + \frac{\left(\frac{x}{2}\right)^4}{2!(n+1)(n+2)} - \dots \right], \quad (53a)$$

а модифицированные функции Бесселя — по разложению

$$I_n(x) = \frac{1}{n!} \left(\frac{x}{2}\right)^n \left[1 + \frac{\left(\frac{x}{2}\right)^2}{1!(n+1)} + \frac{\left(\frac{x}{2}\right)^4}{2!(n+1)(n+2)} + \dots \right]. \quad (53b)$$

Функции Бесселя высших порядков можно вычислять по приводимым в справочниках табличным данным для функций нулевого и первого порядков, воспользовавшись рекуррентной формулой

$$J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x). \quad (53c)$$

Некоторые функции, представляемые определенными интегралами, удобно вычислять по степенному ряду, полученному разложением в ряд Тейлора подынтегрального выражения и почленным интегрированием этого ряда. Например, используя разложение (48а), для интегрального синуса получают ряд

$$\operatorname{Si} x = \int_0^x \frac{\sin u}{u} du = x - \frac{x^3}{3! 3} + \frac{x^5}{5! 5} - \frac{x^7}{7! 7} + \dots, \quad (54)$$

а интеграл вероятности в соответствии с разложением (45а) вычисляют при помощи степенного ряда

$$\Phi(x) = \sqrt{\frac{2}{\pi}} \int_0^x e^{-u^2/2} du = x \sqrt{\frac{2}{\pi}} \left(1 - \frac{(x/2)^2}{1! 3} + \frac{(x/2)^4}{2! 5} - \frac{(x/2)^6}{3! 7} + \dots \right). \quad (55a)$$

Значение заданной буквенной моделью функции иногда удобно вычислять, составив аналитическое выражение для ее первой производной и вычислив интеграл от этого выражения численными методами в пределах от нуля до заданного значения аргумента.

Численное интегрирование при буквенном задании подынтегрального выражения сводится к составлению его табличной модели с $n+1$ равноотстоящими узлами и вычислению интеграла в пределах от $x_0 = 0$ до x_n по формулам (23) или (24).

При больших значениях аргумента эффективность вычислений как с помощью степенного ряда, так и с помощью методов численного интегрирования резко уменьшается. В этом случае иногда удается найти приближение в виде асимптотического ряда, сходящегося к значению заданной функции при $x \rightarrow \infty$. Например, для интеграла вероятности ряд

$$\Phi(x) \approx 1 - \frac{e^{-x^2/2}}{x} \sqrt{\frac{2}{\pi}} \left(1 - \frac{1}{x^2} + \frac{1 \cdot 3}{x^4} - \frac{1 \cdot 3 \cdot 5}{x^6} + \dots \right) \quad (55b)$$

сходится при бесконечном увеличении аргумента. Минимальная ошибка асимптотического разложения стремится к нулю лишь при $x \rightarrow \infty$, а ошибка ограничения не превышает величины последнего оставленного (а не первого отбрасываемого) члена асимптотического ряда.

5. МЕТОДЫ РЕШЕНИЯ УРАВНЕНИЙ

Численное решение уравнения $f(x) = 0$ заключается в определении с заданной точностью такого значения аргумента x_k (называемого нулем или корнем функции), при котором значение функции $f(x_k)$ равно нулю. При табличном задании функции эту задачу решают методами обратного интерполирования, отыскивая корень в интервале между такими двумя смежными узлами модели, в которых значения функции противоположны по знаку.

Решение уравнений по буквенным моделям функций прямыми (точными) методами с помощью формул вычисления корней возможно лишь в некоторых частных случаях. В общем случае для решения таких уравнений используют приближенные численные методы, включающие этап выделения интервалов аргумента, содержащих по одному корню (отделение корней), и этап сокращения этих интервалов (уточнение корней) до размеров, определяемых заданной точностью вычисления корней.

При отделении вещественных корней грубо определяют одну из границ интервала, содержащего корни, и вычисляют значение функции для ряда увеличивающихся значений аргумента. Значение аргумента, при котором изменяется знак функции, равно другой границе интервала, содержащего корень. Если интервал изменения аргумента разбит на ряд отрезков, на границах которых число перемен знака на единицу превышает число корней, то в каждом таком отрезке содержится вещественный корень. Иногда интервал нахождения корня можно определить непосредственно по физическим условиям задачи.

Значения вещественных корней уточняют последовательным разбиением интервала, содержащего корень, на меньшие части до тех пор, пока ширина уменьшаемого интервала, на границах которого значения функции противоположны по знаку, не станет меньше некоторой наперед заданной величины, определяющей точность вычисления корня, требуемую по условию задачи. Точность текущих значений корня удобно оценивать по числу одинаковых значащих цифр в значениях функции на краях сокращаемого интервала.

Вещественные корни в простейшем случае уточняют методом проб с последовательным делением (в частности, пополам) интервала нахождения корня на две части с последующей оценкой знака функции в точке деления и выбором в качестве уточненного интервала той части исходного, на границах которого функция имеет различные знаки. Подобное деление продолжают до тех пор, пока уточняемая длина интервала нахождения корня не станет меньшей некоторой наперед заданной малой величины ϵ , определяющей точность вычисления корня.

Меньшее число приближений требуется при уточнении корня в исходном интервале $[a, b]$ методом хорд по формулам

$$x_{k+1} = x_k - \frac{f(x_k)(b - x_k)}{f(b) - f(x_k)} \text{ или } x_{k+1} = x_k - \frac{f(x_k)(x_k - a)}{f(x_k) - f(a)}, \quad (56)$$

где $k = 0, 1, 2, \dots$, причем выбирается та из приведенных формул, которая соответствует концу интервала a или b (который остается фиксированным при последующих вычислениях) со значением $f(x)f''(x) > 0$.

Погрешность приближения методом хорд оценивают формулой

$$\epsilon = |\xi - x_k| < |x_k - x_{k-1}|,$$

где ξ — точное значение корня. Эта формула справедлива на интервалах, для которых максимальное значение первой производной меньше удвоенного минимального значения первой производной. Если это условие не выполняется, то метод хорд может расходиться.

Достаточно эффективен метод касательных Ньютона, для которого приближенное значение корня уточняется по формуле

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}; \quad k = 0, 1, 2, \dots \quad (57)$$

В качестве исходной точки x_0 выбирают тот конец интервала, в котором знак функции совпадает со знаком второй производной, $f(x)f''(x) > 0$.

Погрешность метода касательных оценивают формулой

$$\epsilon = |\xi - x_k| \leq |f(x_k)| / \min |f'(x)|,$$

справедливой и для метода хорд. Если на исходном интервале максимум второй производной меньше удвоенного минимума первой производной, то точность приближения на k -м шаге оценивают по условию

$$|\xi - x_k| < \epsilon^2, \text{ если } |x_k - x_{k-1}| < \epsilon.$$

Если первая производная мало изменяется в исходном интервале, то в формуле (57) можно принять $f'(x_k) = f'(x_0)$, что существенно упрощает вычисления. В том случае, когда $f'(x)$ и $f''(x)$ сохраняют знак в исходном интервале, методы хорд и касательных можно объединить, построив вычислительный процесс в соответствии с формулами

$$\bar{x}_k = \bar{x}_{k-1} - \frac{f(\bar{x}_{k-1})}{f'(\bar{x}_{k-1})}; \quad x_{k+1} = -\frac{f(x_k)(\bar{x}_k - x_k)}{f(\bar{x}_k) - f(x_k)} + x_k; \quad k = 1, 2, \dots \quad (58)$$

Точное значение корня в этом случае лежит в интервале (x_k, \bar{x}_k) и, если достигнут отрезок $|\bar{x}_k - x_k| < \epsilon$, то выбирают $\xi = 0,5(x_k + \bar{x}_k)$. В этом методе, в отличие от описываемых формулами (56) и (57), совпадающие значащие цифры \bar{x}_k и x_k принадлежат точному значению корня ξ .

Часто для решения уравнений используют метод последовательных приближений (простых итераций), при котором $f(x) = 0$ заменяют равносильным уравнением $x = \varphi(x)$ и вычисляют сходящуюся к корню последовательность чисел

$$x_k = \varphi(x_{k-1}); \quad k = 1, 2, \dots \quad (59)$$

Если функция в правой части уравнения определена и дифференцируема на исходном участке, то при $|\varphi'(x)| \leq q < 1$, где q — правильная дробь, процесс итераций (59) сходится независимо от выбора начального приближения и погрешностей промежуточных вычислений.

Приближенную оценку точности результата при этом методе дает формула

$$|\xi - x_k| \leq \frac{q^k}{1-q} |x_1 - x_0|.$$

При $q < 0,5$ из этого неравенства следует $|\xi - x_k| \leq |x_k - x_{k-1}|$. Необходимо подчеркнуть, что в общем случае совпадение значащих цифр x_k и x_{k-1} при итерациях по формуле (59) не означает, что значение корня определено с той же точностью.

Комплексный корень аналитической функции можно вычислить по формуле (57). Более общий метод основан на представлении функции $f(x) = u(\sigma, \omega) + jv(\sigma, \omega)$ комплексного аргумента $x = \sigma + j\omega$ системой уравнений $u(\sigma, \omega) = 0, v(\sigma, \omega) = 0$ для функций вещественных аргументов σ и ω , корни которых находят методами решения систем уравнений.

Алгебраическое уравнение m -й степени

$$P(x) \equiv a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0 = 0 \quad (60)$$

с вещественными коэффициентами имеет m вещественных или комплексно-сопряженных корней (нулей), причем для нечетного m веществен по крайней мере один корень.

Область расположения корней уравнения (60) на плоскости комплексного аргумента ограничена круговым кольцом $r < |x| < R$, где $R = 1 + |a_{\max}| / |a_m|$ и $r = 1 / (1 + |a_{\max-1}| / |a_m|)$, a_{\max} — наибольший по модулю коэффициент многочлена, а $a_{\max-1}$ — наибольший по модулю коэффициент без учета a_{\max} . Для положительных вещественных корней более точная верхняя граница определяется числом $R = 1 + (b/a_m)^{1/k}$, где b — модуль наибольшего отрицательного коэффициента, k — индекс первого из отрицательных коэффициентов (если все коэффициенты положительны, то многочлен не имеет положительных корней). Нижняя граница положительных корней определяется по формуле $r = 1/R$ после подстановки $x = 1/y$. Границы отрицательных корней определяются аналогично после подстановки $x = -z$.

Число положительных (отрицательных) корней алгебраического многочлена равно числу перемен (постоянств) знаков в последовательности a_m, \dots, a_0 коэффициентов или меньше его на четное число.

Корни алгебраических уравнений степени $m \leq 4$ можно определить точными методами за конечное число шагов [14]. Корни многочлена второй степени $a_2 x^2 + a_1 x + a_0$ вычисляют по формуле

$$x_{1,2} = (-a_1 \pm \sqrt{a_1^2 - 4a_0 a_2}) / 2a_2. \quad (61)$$

Вследствие ошибок округления корни многочленов степени 3 и 4, вычисленные точными методами, могут иметь значительную погрешность. Поэтому при $m \geq 3$ для решения алгебраических уравнений используют приближенные методы, основанные, например, на выделении вещественного корня многочленов нечетной степени и раз-

ложении многочленов четной степени на произведения сомножителей вида $x^2 + px + q$.

Для выделения квадратичных сомножителей многочлен четной степени представляют тождеством

$$P(x) = (x^2 + px + q)P_1(x) + xP(p, q) + Q(p, q), \quad (62)$$

где $P_1(x)$ — частное от деления $P(x)$ на квадратичный трехчлен; $P(p, q)$ и $Q(p, q)$ — многочлены от p и q в остатке.

Многочлен $P_1(x)$ является точным множителем $P(x)$, если $P(p, q) = 0$ и $Q(p, q) = 0$. Решив эту систему уравнений относительно искомых коэффициентов p и q , находят корни выделенного трехчлена по формуле (61) и продолжают разложение $P_1(x)$ до выделения всех множителей второй степени и вычисления их корней.

При $m > 4$ использование формулы (62) связано с большими затратами времени, что заставляет применять другие методы вычисления корней многочленов высокой степени. К ним относится метод Лобачевского — Греффе, позволяющий разделить исходное уравнение на несколько более простых. Это достигается k -кратным преобразованием (квадрированием) коэффициентов a_i исходного многочлена, приводящим к уравнению

$$y^m + A_{m-1, k} y^{m-1} + \dots + A_{1, k} y + A_{0, k} = 0, \quad (63a)$$

где $y = x^{2^k}$, а коэффициенты этого уравнения при каждом $(j+1)$ -м квадрировании вычисляют по формуле

$$A_{i, i+1} = A_{ij}^2 + \sum_{s=1}^v 2(-1)^s A_{(i-s)j} A_{(i+s)j}, \quad (63b)$$

где v — меньшее из чисел i и $m-1$.

Квадрирование увеличивает различие между корнями преобразованного уравнения в α^{2k} раз, где α — отношение модулей корней исходного уравнения. Например, если $\alpha = 1,2$, то при $k = 6$ получим $\alpha^{2k} = 1,2^{64} \approx 10^5$. Обычно достаточно выполнить 5—6 квадрирований.

Разделение корней контролируется по соотношениям между $A_{i, i+1}$ и A_{ij} . Условия разделения выполняются, если с требуемой точностью выполняется равенство $A_{i, i+1} = A_{ij}^2$. Корни с равными модулями не разделяются, в связи с чем при $m \leq 6$ возможны следующие варианты изменения коэффициентов и значения корней:

1. Если условия разделения выполняются для всех коэффициентов, то корни вещественные и различные, а их модули определяются формулой

$$|x_i| = (A_{ik}/A_{i-1, k})^{1/2^k}. \quad (64a)$$

2. Если для несмежных коэффициентов условия разделения не выполняются и после каждого квадрирования они изменяют знак (флюктуируют), то имеются комплексно-сопряженные корни. Если

флюктуирует один коэффициент, то модуль пары комплексно-сопряженных корней определяется формулой

$$|x_i| = (A_{i+1,k}/A_{i-1,k})^{(1/2)(1/2^k)}. \quad (64)$$

Для определения вещественной и мнимой частей этой пары корней используют соотношение $a_{m-1} = x_1 + x_2 + \dots + 2 \operatorname{Re} x_m$, позволяющее после определения всех $m-2$ вещественных корней вычислить $\operatorname{Re} x_m$, а затем $\operatorname{Im} x_m = [|x_m|^2 - (\operatorname{Re} x_m)^2]^{1/2}$.

При наличии нескольких комплексно-сопряженных пар корней их модули вычисляют по формуле (64), а для определения вещественных и мнимых частей используют уравнения с большим числом коэффициентов исходного уравнения.

3. Если для одного из коэффициентов условия разделения не выполняются, причем квадрированный коэффициент стремится к половине квадрата предыдущего $(j-1)$ -го значения, то имеется пара вещественных кратных или противоположных по знаку корней с модулем (64).

4. Если два смежных коэффициента A_{ik} и $A_{i+1,k}$ не разделяются, то имеется вещественный корень и пара комплексно-сопряженных корней с одинаковыми модулями

$$|x| = (A_{i+2,k}/A_{i-1,k})^{(1/3)(1/2^k)}.$$

5. Если не разделяются три смежных коэффициента A_i , A_{i+1} и A_{i+2} , то имеются два вещественных и пара комплексно-сопряженных корней с модулями

$$|x| = (A_{i+3,k}/A_{i-1,k})^{(1/4)(1/2^k)}.$$

В двух последних случаях вещественные и мнимые части комплексно-сопряженных пар корней определяются как во втором. Правильность выбора знака и значений вещественной и мнимой частей корня определяется его подстановкой в исходное уравнение.

Алгебраический многочлен (27) называют устойчивым, если все корни уравнения (60) имеют отрицательные вещественные части. При решении ряда инженерных задач возникает необходимость в проверке устойчивости алгебраических многочленов. Подобную проверку можно выполнить без вычисления корней многочлена, используя критерии его устойчивости.

Согласно критерию Рауса — Гурвица алгебраический многочлен $P(x)$ степени m устойчив, если

$$a_m > 0; a_{m-1} > 0; \begin{vmatrix} a_{m-1} & a_m \\ a_{m-3} & a_{m-2} \end{vmatrix} > 0; \dots; \begin{vmatrix} a_{m-1} & a_m & 0 & \dots \\ a_{m-3} & a_{m-2} & a_{m-1} & \dots \\ \dots & \dots & \dots & \dots \\ \dots & 0 & a_0 & a_1 \end{vmatrix} > 0; \\ a_0 > 0. \quad (65a)$$

При составлении входящих в эти неравенства определителей следует учесть, что на главной диагонали определителя высшего порядка расположены коэффициенты $a_{m-1}, a_{m-2}, \dots, a_1$, а в каждой строке — смежные коэффициенты по возрастающим степеням. Определители низших порядков образуются последовательным вычеркиванием нижней строки и правого столбца.

При оценке устойчивости многочленов с большой степенью m удобно использовать критерий Гурвица, основанный на разложении отношения четной $M(x) = \dots + a_2x^2 + a_0$ к нечетной $N(x) = \dots + a_3x^3 + a_1x$ частей многочлена четной степени или обратного отношения для многочлена нечетной степени в обыкновенную цепную дробь

$$C(x) = c_1x + \frac{1}{c_2x + \frac{1}{c_3x + \dots + \frac{1}{c_mx}}}.$$

Согласно этому критерию многочлен устойчив, если коэффициенты этой цепной дроби удовлетворяют условию

$$c_k > 0; k = 1, 2, \dots, m. \quad (65)$$

В инженерной практике иногда используют частотные критерии, основанные на подстановке $x = j\omega$ в $P(x) = P(j\omega)$ с последующим построением частотных годографов* функций $F(j\omega) = P(j\omega)/P_0(j\omega)$ или $T(j\omega) = F(j\omega) - 1 = [P(j\omega) - P_0(j\omega)]/P_0(j\omega)$, где $P_0(j\omega)$ — заведомо устойчивый многочлен, полученный из $P(j\omega)$ соответствующим изменением коэффициентов.

Согласно обобщенному критерию Найквиста многочлен $P(x)$ устойчив, если частотный годограф функции $F(j\omega)$ или $T(j\omega)$ не охватывает критической точки с координатами $(0, 0)$ или $(-1, 0)$ соответственно. Под словом «охватывает» подразумевается, что если при увеличении частоты ω от $-\infty$ до ∞ (части годографа при $\omega > 0$ и $\omega < 0$ симметричны) образуется замкнутый контур, то критическая точка остается справа (по отношению к направлению возрастания частоты) от годографа. Обозначив область справа от годографа символом Π , можно отобразить этот критерий формулой

$$(0, 0) \notin \Pi F(j\omega); (-1, 0) \notin \Pi T(j\omega). \quad (65b)$$

По кратчайшему расстоянию от критической точки до годографа устойчивого многочлена судят о запасе устойчивости по отношению к заведомо устойчивому многочлену $P_0(x)$. Если в этом нет необходимости, то для упрощения вычислений целесообразно принять $P_0(j\omega) = 1$.

* Частотным годографом функции $f(j\omega)$ называют геометрическое место точек функции на плоскости с координатами $\operatorname{Re} f(\omega)$ и $\operatorname{Im} f(\omega)$.

Обыкновенное дифференциальное уравнение r -го порядка

$$F(x, y(x), y'(x), \dots, y^{(r)}(x)) = 0 \quad (66a)$$

связывает независимую переменную x , искомую функцию $y(x)$ и ее производные $y'(x), \dots, y^{(r)}(x)$ по аргументу. Решение (интегрирование) дифференциального уравнения (66а) заключается в отыскании значений $y(x)$, удовлетворяющих уравнению для всех значений x в заданном конечном или бесконечном интервале. Результат решения (66а) имеет вид $y = y(x, C_1, \dots, C_r)$, где постоянные C_1, \dots, C_r определяются начальными условиями $y(x_0) = y_0, y'(x_0) = y'_0, \dots, y^{(r-1)}(x_0) = y^{(r-1)}_0$.

Решение дифференциального уравнения (66а) сводят к решению системы дифференциальных уравнений первого порядка заменой высших производных неизвестными функциями

$$y' = z_1; z'_1 = z_2; \dots; z'_r = f(x, y, z_1, \dots, z_r). \quad (66b)$$

Решение такой системы уравнений заключается в последовательном решении дифференциальных уравнений первого порядка.

Точные аналитические решения дифференциальных уравнений первого порядка существуют для ограниченного числа таких уравнений. Поэтому обычно их решают (интегрируют) приближенными аналитическими или численными методами.

К приближенным аналитическим методам интегрирования относятся методы последовательного дифференцирования и последовательных приближений. Первый из них основан на вычислении высших производных правой части (аналитической в интервале аргумента) дифференциального уравнения первого порядка

$$\frac{dy}{dx} \equiv y' = f(x, y) \quad (67)$$

при начальных условиях $x = x_0, y(x_0) = y_0$. Подставив значения производных при $x = x_0$ в ряд Тейлора (32), получают степенной ряд, позволяющий вычислить значение $y(x)$ тем точнее, чем меньше разность $x - x_0$. Этот метод непосредственно применим и для тех уравнений высокого порядка, для которых удается вычислить младшие производные, входящие в члены аппроксимирующего ряда.

Второй метод (последовательных) приближений основан на использовании рекуррентной формулы

$$y_k(x) = y_0 + \int_{x_0}^x f(x, y_{k-1}) dx; \quad k = 1, 2, \dots \quad (68)$$

при выборе в качестве первого приближения

$$y_1(x) = y_0 + \int_{x_0}^x f(x, y_0) dx.$$

Обычно используются методы численного интегрирования дифференциальных уравнений первого порядка. Они разбиваются на одноступенчатые и многостепенные. В первых очередное значение функции y_{i+1} вычисляют по предыдущему ее значению y_i , тогда как во вторых — по некоторым предыдущим значениям. Многостепенные методы обеспечивают большую точность решения, но относительно громоздки, а в начале вычислений и при изменении шага интегрирования несколько значений функции приходится вычислять одноступенчатыми методами.

Большинство одноступенчатых методов относится к методам Рунге — Кутта различных порядков. К ним, в частности, относятся классический метод Эйлера, описываемый формулой

$$y_{i+1} = y_i + hy'_i; \quad h = x_{i+1} - x_i; \quad i = 0, 1, 2, \dots, \quad (69a)$$

модифицированный метод Эйлера

$$y_{i+1} = y_i + hy'(x_i + h/2, y_i + hy'_i/2); \quad i = 0, 1, 2, \dots, \quad (69b)$$

усовершенствованный метод Эйлера — Коши

$$y_{i+1} = y_i + [y'_i + y'(x_{i+1}, y_i + hy'_i)]h/2; \quad i = 0, 1, 2, \dots, \quad (69b)$$

а также метод Рунге — Кутта четвертого порядка (часто называемый без указания порядка методом Рунге — Кутта)

$$y_{i+1} = y_i + \frac{1}{6}(g_1 + 2g_2 + 2g_3 + g_4); \quad i = 0, 1, 2, \dots, \quad (69c)$$

где $g_1 = hy'(x_i, y_i)$; $g_2 = hy'(x_i + h/2, y_i + g_1/2)$; $g_3 = hy'(x_i + h/2, y_i + g_2/2)$; $g_4 = hy'(x_{i+1}, y_i + g_3)$.

К многостепенным относятся уточненный метод Эйлера

$$y_{i+1} = y_{i-1} + 2hy'_i; \quad i = 2, 3, \dots, \quad (70)$$

где y_1 вычисляют по формуле (69a), а последующие значения функции y_i определяют по формуле $y_i = y_{i-1} + hy'_{i-1}; \quad y_{1/2} = y_{i-1} + hy'_{i-1}/2$, а также экстраполяционно-итерационный метод Адамса и различные методы прогноза и коррекции.

При достаточно малом шаге $h = x_{i+1} - x_i$ и известных предыдущих четырех значениях функции ее очередное значение в схеме Адамса определяется формулой

$$y_{i+1} = y_i + q_i + \frac{1}{2} \Delta q_{i-1} + \frac{5}{12} \Delta^2 q_{i-2} + \frac{3}{8} \Delta^3 q_{i-3}, \quad (71a)$$

где $\Delta^m q_i$ — конечные m -е разности $q_i = hy'_i$. Для вычисления первых трех значений функции целесообразно использовать (при известном нулевом приближении) формулы Крылова

$$\begin{aligned} \Delta y_0 &= q_0 + \frac{1}{2} \Delta q_0 - \frac{1}{12} \Delta^2 q_0 + \frac{1}{24} \Delta^3 q_0; \\ \Delta y_1 &= q_1 + \frac{1}{2} \Delta q_0 + \frac{5}{12} \Delta^2 q_0 - \frac{1}{24} \Delta^3 q_0; \\ \Delta y_2 &= q_2 + \frac{1}{2} \Delta q_1 + \frac{5}{12} \Delta^2 q_1 + \frac{3}{8} \Delta^3 q_0. \end{aligned} \quad (71b)$$

В методе прогноза и коррекции Милна для получения первого приближения y'_{i+1} используется формула прогноза

$$\bar{y}_{i+1} = y_{i-3} + \frac{4h}{3} (2y'_i - y'_{i-1} + 2y'_{i-2}), \quad (72a)$$

после чего вычисляют корректированное значение

$$y_{i+1} = y_{i-1} + \frac{h}{3} (\bar{y}'_{i+1} + 4y'_i + y'_{i-1}). \quad (72b)$$

Вычисления производят при таком шаге h , для которого значения функции, вычисленные по формулам (72), отличаются не более чем на 14 единиц последнего сохраняемого десятичного разряда.

Начальные значения функции в этом методе могут быть вычислены по формулам Крылова (71б) или формулам Милна:

$$\begin{aligned} y_{-1} &= y_0 - \frac{h}{24} (y'_1 + 16y'_0 + 7y'_{-1}) + \frac{h^2}{4} y''(x_0); \\ y_1 &= y_0 + \frac{h}{24} (7y'_1 + 16y'_0 + y'_{-1}) + \frac{h^2}{4} y''(x_0); \\ y_2 &= y_0 + \frac{2h}{3} (5y'_1 - y'_0 - y'_{-1}) - 2h^2 y''(x_0) \end{aligned}$$

с корректировкой вычисленных значений согласно формулам (72б).

При решении дифференциальных уравнений рассмотренными методами начальные значения должны быть вычислены особенно точно, так как их погрешности накапливаются в последующих вычислениях.

Реакция (4) инерционного канала причинно-следственной связи на воздействия $x(t)$ и $z(t)$ определяется решением уравнения (6б) с независимой переменной t :

$$F(t, y(t), y'(t), \dots, y^{(r)}(t)) = 0. \quad (73a)$$

Линейные инерционные каналы связи в общем случае (при $z = \text{const}$) описываются обыкновенным дифференциальным уравнением вида

$$c_r \frac{d^r y}{dt^r} + \dots + c_1 \frac{dy}{dt} + c_0 y = b_m \frac{d^m x}{dt^m} + \dots + b_1 \frac{dx}{dt} + b_0 x. \quad (73b)$$

Применяя к обеим частям такого уравнения преобразование Лапласа

$$F(p) = \int_0^\infty f(t) e^{-pt} dt, \quad (74)$$

получают алгебраическое (операторное) уравнение (6а) для вычисления $y(p)$. Изображение $y(p)$ отображается в область аргумента t с помощью обратного преобразования Лапласа, приближенно реализуемого различными косвенными методами [14].

Производные в дифференциальных уравнениях при достаточно малом шаге Δt аргумента приближенно выражаются через конеч-

ные разности с помощью формул (22а). Это соответствует аппроксимации дифференциального уравнения разностным уравнением

$$G(t_i, y_i, y_{i+1}, \dots, y_{i+r}) = 0; \quad i = 0, \pm 1, \pm 2, \dots, \quad (75a)$$

связывающим значения $y_i = y(t_i)$ на дискретном множестве значений аргумента $t_i = t_0 + i \Delta t$. В частности, линейное дифференциальное уравнение (73б) аппроксимируется неоднородным разностным уравнением

$$\omega_0 y_i + \omega_1 y_{i-1} + \dots + \omega_r y_{i-r} = v_0 x_i + v_1 x_{i-1} + \dots + v_m x_{i-m}, \quad (75b)$$

связывающим последовательности значений $x_i = x(t_i) = x(i)$ и $y_i = y(x(t_i)) = y(i)$.

При операциях с дискретными последовательностями $f(i)$ часто используют прямое и обратное z -преобразования

$$f(z) = \sum_{i=0}^{\infty} f(i) z^{-i}; \quad f(i) = \frac{1}{2\pi j} \oint f(z) z^{i-1} dz, \quad (76a)$$

где $z = e^{p\theta}$ — комплексная переменная, а $f(z)$ — функция этой переменной, называемая изображением функции-оригинала $f(i)$.

Если функцию $f(z)$ представить в виде ряда

$$f(z) = \sum_{k=0}^{\infty} c_k z^{-k}, \quad (76b)$$

то ее оригиналом будет

$$f(i) = \sum_{k=0}^{\infty} c_k \delta(k - i), \quad (76b)$$

где $\delta(i)$ — единичный импульс, определяемый как

$$\delta(i) = \begin{cases} 1 & \text{при } i = 0; \\ 0 & \text{при } i \neq 0. \end{cases}$$

Соотношение (76в) непосредственно вытекает из свойств z -преобразования.

При вычислениях по разностному уравнению (75б) его обычно приводят к виду

$$y(i) = \sum_{k=0}^M \frac{v_k}{w_0} x(i-k) - \sum_{k=1}^N \frac{w_k}{w_0} y(i-k); \quad i = 0, 1, 2, \dots, \quad (77)$$

где $y(i-k) = x(i-k) = 0$ при $k > i$.

При цифровом моделировании поведения аналоговых (непрерывных) систем может возникнуть задача определения коэффициентов уравнения (77) по заданной функции моделируемой системы $a(p)$. Среди различных способов определения коэффициентов этого разностного уравнения практический интерес представляет преоб-

разование уравнения вида (6а), основанное на подстановке в дробно-рациональную функцию (6б) комплексной переменной z согласно соотношению

$$p = (z - 1)/(z + 1) \quad (78a)$$

с последующим делением числителя и знаменателя функции $a(z)$ на старшую степень z . В этом случае получают

$$a(z) = \frac{y(z)}{x(z)} = \frac{v_0 + v_1 z^{-1} + \dots + v_m z^{-m}}{w_0 + w_1 z^{-1} + \dots + w_m z^{-m}}, \quad (78b)$$

откуда находят

$$y(z) [w_0 + \dots + w_m z^{-m}] = x(z) [v_0 + \dots + v_m z^{-m}]$$

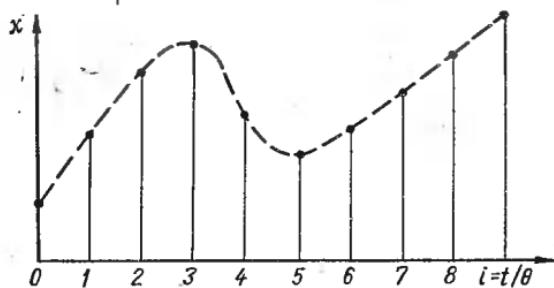


Рис. 6. Представление кусочно-непрерывной функции $x(t)$ последовательностью дискретных значений x_i с шагом дискретизации θ

следующих через равные промежутки времени $\theta = \Delta t$. Воздействие $x(t)$, описываемое кусочно-непрерывной функцией, также можно представить (рис. 6) последовательностью таких импульсов, но погрешность подобного представления будет тем большей, чем больше шаг дискретизации θ .

Реакция (4) линейного параметрического канала является решением линейного дифференциального уравнения с переменными параметрами, а при различных координатах отсчета воздействий x и z — решением дифференциальных уравнений с частными производными [14].

6. МЕТОДЫ РЕШЕНИЯ СИСТЕМ УРАВНЕНИЙ

В инженерной практике часто приходится находить численные или буквенные (символические) решения систем линейных уравнений вида

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= q_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= q_2; \\ &\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= q_n, \end{aligned} \quad (79a)$$

или, с учетом соотношения (78б), разностное уравнение (77), позволяющее вычислять отсчеты $y_i = y(t_i)$ по отсчету $x_i = x(t_i)$ и предыдущим отсчетам y_{i-k} и x_{i-k} .

Вычисления по формуле (77) дают точные результаты в тех случаях, когда воздействие $x(t)$ представляет собой последовательность достаточно коротких импульсов $x(t_i)$,

следующих через равные промежутки времени $\theta = \Delta t$. Воздействие $x(t)$, описываемое кусочно-непрерывной функцией, также можно представить (рис. 6) последовательностью таких импульсов, но погрешность подобного представления будет тем большей, чем больше шаг дискретизации θ .

Реакция (4) линейного параметрического канала является решением линейного дифференциального уравнения с переменными параметрами, а при различных координатах отсчета воздействий x и z — решением дифференциальных уравнений с частными производными [14].

записываемых в виде матричного уравнения

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{bmatrix} \quad (79)$$

или в сокращенной символической (операторной) форме записи

$$AX = Q. \quad (79\text{в})$$

Задача численного решения такой системы, моделирующей некоторый физический объект или ситуацию, сводится к нахождению значений x_i по известным значениям элементов матрицы A и вектора-столбца Q . При буквенном решении составляют аналитические выражения искомых переменных x_i через буквенные символы коэффициентов (параметров модели) a_{ij} и свободных членов (независимых переменных воздействий) q_i .

Система линейных уравнений (79) имеет единственное решение $X = A^{-1}Q$, если определитель Δ матрицы A не равен нулю (матрица A не особенная). В этом случае неизвестные значения x_i определяются по правилу Крамера соотношениями

$$x_j = \sum_{i=1}^n f_{ji} q_i = \sum \Delta_{ij} q_i / \Delta = \Delta_j / \Delta; \quad j = 1, 2, \dots, n, \quad (80\text{a})$$

где f_{ji} — элементы обратной матрицы $F = A^{-1}$; Δ_{ij} — алгебраические дополнения элементов a_{ij} матрицы A ; Δ — определитель матрицы A , в которой j -й столбец заменен столбцом свободных членов q_i .

Буквенное решение системы уравнений (79), моделирующей согласно формуле (8а) физическую систему со слабыми воздействиями, качественно характеризует зависимость искомых переменных реакций от параметров модели a_{ij} и переменных воздействий q_i . Поэтому при моделировании элементы обратной матрицы

$$f_{ji} = x_j / q_i = \Delta_{ij} / \Delta \quad (80\text{б})$$

называют входными (при $i = j$) или передаточными (при $i \neq j$) функциями или вторичными параметрами модели. Отношение между переменными реакции часто определяют функциями вида

$$k_{ji} = x_j / x_i = f_{ji} / f_{ii} = \Delta_{ij} / \Delta_{ii} \quad (80\text{в})$$

в предположении о независимости переменной x_i .

Численные значения x_i можно найти в соответствии с формулами (80) после обращения матрицы A и вычисления элементов обратной матрицы. Однако такой подход связан со значительными затратами времени и обычно для численного решения системы линейных уравнений используют точные (прямые) или приближенные (итерационные или шаговые) методы.

Прямые методы основаны на последовательном численном преобразовании заданной системы уравнений по известным формулам за конечное число шагов к виду, позволяющему непосредственно найти численные значения переменных x_i . В этом случае остаточная погрешность отсутствует и точность результатов определяется лишь ошибками округления в процессе вычислений.

Среди прямых методов наиболее часто используют метод исключения Гаусса и его различные модификации. В наиболее распространенном варианте этого метода, называемом схемой единственного деления, уравнения системы (79а) предварительно переставляются так, чтобы все диагональные элементы матрицы A возможно больше отличались от нуля.

Операции в схеме единственного деления производятся над элементами расширенной матрицы $[A, Q]$ размера $n \times (n + 1)$, отличающейся от матрицы A дополнительным столбцом свободных членов $q_i = a_{i,n+1}$. Схема разбивается на прямой и обратный ходы. На каждом p -м шаге ($p = 1, 2, \dots, n$) прямого хода вычисляются новые значения элементов p -й строки

$$a_{pj}^{(p)} = a_{pj}^{(p-1)} / a_{pp}^{(p-1)}; \quad j > p, \quad (81a)$$

где a_{pp} называется ведущим элементом, и последующих строк

$$a_{ij}^{(p)} = a_{ij}^{(p-1)} - a_{ip}^{(p-1)} a_{pj}^{(p)}; \quad i > p; \quad j > p. \quad (81b)$$

В результате n шагов прямого хода исходная система уравнений преобразуется в систему уравнений с верхней треугольной матрицей вида

$$\begin{bmatrix} 1 & u_{12} & \dots & u_{1n} \\ 0 & 1 & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} u_{1,n+1} \\ u_{2,n+1} \\ \dots \\ u_{n,n+1} \end{bmatrix} \quad (81b)$$

с известным значением $x_n = u_{n,n+1}$. Значения остальных неизвестных вычисляют по уравнениям системы (81b) согласно формулам обратного хода

$$x_p = u_{p,n+1} - \sum_{k=p+1}^n u_{pk} x_k; \quad p = n-1, n-2, \dots, 1. \quad (81c)$$

Определитель матрицы A равен произведению ведущих элементов в схеме единственного деления:

$$\Delta = a_{11} a_{22}^{(1)} \dots a_{nn}^{(n-1)}. \quad (81d)$$

Один из часто используемых вариантов метода исключений связан с разложением исходной системы уравнений $AX = Q$ на две системы $LY = Q$ и $UX = Y$, где L — нижняя и U — верхняя треугольные матрицы с единичными диагональными элементами. Подобное LU -разложение реализуется по компактной схеме (схеме Халецкого), при прямом ходе которой элементы расширенной

матрицы $[A, Q]$ заменяются элементами матриц L и U . В результате прямого хода с вычислением по формулам

$$l_{11} = a_{11}; \quad l_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, \quad i \geq j > 1; \quad (82a)$$

$$u_{1j} = a_{1j}/l_{11}; \quad u_{ij} = \frac{1}{l_{11}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \right); \quad j > i > 1 \quad (82b)$$

исходная система уравнений преобразуется в систему

$$\begin{bmatrix} l_{11} & u_{12} & \dots & u_{1n} \\ l_{21} & l_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} u_{1,n+1} \\ u_{2,n+2} \\ \dots \\ u_{n,n+1} \end{bmatrix}. \quad (82c)$$

Искомые значения x_i вычисляют в процессе обратного хода через вспомогательные переменные y_i по формулам

$$y_1 = \frac{u_{1,n+1}}{l_{11}}; \quad y_i = \frac{1}{l_{11}} \left(u_{i,n+1} - \sum_{k=1}^{i-1} l_{ik} y_k \right); \quad i > 1; \quad (82d)$$

$$x_n = y_n; \quad x_i = y_i - \sum_{k=i+1}^n u_{ik} x_k; \quad i < n. \quad (82e)$$

Если матрица A симметрическая ($a_{ij} = a_{ji}$), то $u_{ij} = l_{ji}/l_{ii}$ и вычисления существенно упрощаются.

Определитель матрицы A вычисляется по результатам LU -разложения как

$$\Delta = l_{11} l_{22} \dots l_{nn}. \quad (82f)$$

Некоторые варианты метода исключений позволяют вычислить искомые переменные в результате прямого хода. К ним относится метод Гаусса — Жордана, основанный на преобразовании расширенной матрицы $[A, Q]$ в матрицу $[I, X]$, при котором элементы матрицы A последовательно заменяются элементами единичной матрицы, а элементы вектора Q — элементами вектора X .

Реализация метода Гаусса — Жордана (являющегося частным случаем рассматриваемого метода) подобна схеме единственного деления, но на каждом p -м шаге по формулам (81a) и (81b) преобразуют не только элементы строк с номерами $i \geq p$, но и строк с номерами $i < p$. В результате n шагов прямого хода, на каждом из которых получают столбец (в котором только один элемент равен единице, а остальные равны нулю), расширенная матрица преобразуется в матрицу

$$[IX] = \begin{bmatrix} 1 & 0 & \dots & 0 & x_1 \\ 0 & 1 & \dots & 0 & x_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & x_n \end{bmatrix} \quad (83)$$

со столбцом искомых значений x_i .

В рассмотренных методах влияние погрешностей округления резко возрастает при близости к нулю значений ведущих элементов главной диагонали матрицы. Эти погрешности можно уменьшить, используя метод главных элементов. В этом методе на каждом p -м шаге выбирают наибольший по модулю элемент $a_{r_p s_p}$, называемый главным, и вычисляют новые значения неглавных строк по формуле

$$a_{ij}^{(p+1)} = a_{ij}^{(p)} - a_{is_p}^{(p)} / a_{r_p s_p}^{(p)}. \quad (84)$$

Вычеркивая главные r_p -ю строку и s_p -й столбец, в полученной матрице M_{p+1} выбирают главный элемент и продолжают подобные преобразования до получения матрицы M_n с одной строкой, принимаемой за главную. Объединяя все полученные главные строки, получают систему уравнений с треугольной матрицей, по которой в процессе обратного хода вычисляют искомые значения x_i .

Приближенные (итерационные) методы решения системы линейных уравнений основаны на выборе некоторого начального приближения с последующим их уточнением по рекуррентным формулам до получения результата с заданной точностью.

При использовании метода итераций (последовательных приближений) систему линейных уравнений (79) предварительно преобразуют делением каждого i -го уравнения к виду

$$\begin{aligned} x_1 &= \beta_1 + \alpha_{12}x_2 + \dots + \alpha_{1n}x_n; \\ x_2 &= \beta_2 + \alpha_{21}x_1 + \dots + \alpha_{2n}x_n; \\ &\dots \dots \dots \dots \dots \dots \\ x_n &= \beta_n + \alpha_{n1}x_1 + \dots + \alpha_{n, n-1}x_{n-1}. \end{aligned} \quad (85)$$

Принимая начальные приближения x_i (например, $x_i^{(0)} = \beta_i$), подставляют их в правую часть системы уравнений (85) и вычисляют значения $x_i^{(1)}$ левой части, которые снова подставляют в правую часть и по ним вычисляют значения $x_i^{(2)}$ в левой части. Подобные итерации повторяют до выполнения условия $|x_i^{(k+1)} - x_i^{(k)}| \leq \varepsilon$, где ε — число, определяющее заданную точность вычисления неизвестных.

В модифицированном методе последовательных приближений (методе Зейделя) исходная система уравнений (79) преобразуется к виду

$$\begin{aligned} x_1 &= \beta_1 + \alpha_{11}x_1 + \alpha_{12}x_2 + \dots + \alpha_{1n}x_n; \\ x_2 &= \beta_2 + \alpha_{21}x_1 + \alpha_{22}x_2 + \dots + \alpha_{2n}x_n; \\ &\dots \dots \dots \dots \dots \dots \\ x_n &= \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + \dots + \alpha_{nn}x_n. \end{aligned} \quad (86)$$

Выбирая начальные приближения $x_i^{(0)}$, подставляют их в систему (86) и вычисляют значение $x_i^{(1)}$, которое подставляют в остальные уравнения и вычисляют $x_2^{(1)}$ и т. д. После вычисления $x_n^{(1)}$ под-

ставляют его в правую часть, вычисляют $x_i^{(2)}$ и снова повторяют процесс до тех пор, пока различие между очередными значениями $x_i^{(k+1)}$ и $x_i^{(k)}$ не будет меньше наперед заданного числа ϵ .

Условия сходимости методов последовательных приближений определяют по нормам матрицы (или вектора) A :

$$\|A\|_m = \max_i \sum_{j=1}^n |a_{ij}|; \quad (87a)$$

$$\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|; \quad (87b)$$

$$\|A\|_k = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2}. \quad (87c)$$

Рассмотренные итерационные процессы сходятся к единственному решению, если хотя бы одна из норм матрицы A исходной системы уравнения (79) меньше единицы. Систему линейных уравнений посредством элементарных преобразований всегда можно привести к виду, удовлетворяющему условию сходимости методов последовательных приближений.

Если задана допустимая абсолютная погрешность ϵ , то погрешность $|X - X^{(k)}| \leq \epsilon$ процесса итераций оценивают по формуле

$$\|X - X^{(k)}\| \leq \frac{\|\alpha\|^{k+1}}{1 - \|\alpha\|} \|\beta\|,$$

где $\|\alpha\|$ и $\|\beta\|$ — одна из норм (87) матрицы α и вектора β ; k — число итераций, обеспечивающее заданную точность.

Погрешность метода Зейделя оценивают по формуле

$$\|X - X^{(k)}\|_m \leq \frac{\|\alpha\|_m^k}{1 - \|\alpha\|_m} \|X^{(1)} - X^{(0)}\|,$$

где индексом m обозначена m -норма (87a).

При точном решении системы линейных уравнений их невязки

$$\delta_i \equiv \sum_{j=1}^n a_{ij}x_j - q_i; \quad i = 1, 2, \dots, n \quad (88)$$

равны нулю. Точность вычисления вектора X оценивают по нормам (87) вектора невязок δ . В тех случаях, когда выбранная для оценки норма невязок велика, полученное решение системы уравнений можно уточнить, представив его системой $A(X + \sigma) = 0$, где σ — вектор отклонения переменных x_i от точных значений, связанный с вектором невязок отношением $A\sigma = \delta$. Следовательно, для вычисления поправок σ_i к вычисленным значениям x_i необходимо решить исходную систему уравнений, заменив вектор Q свободных членов вектором невязок δ .

Во многих случаях удобны релаксационные методы решения систем линейных уравнений, при которых для выбранных начальных значений вычисляют невязки δ_i и следующие приближения выбирают при равенстве нулю некоторых невязок. Различают следующие релаксационные методы и их комбинации [14]:

1. Координатная релаксация — на каждом шагу устраниют наибольшую невязку, вычисляя поправку соответствующей переменной x_i .

2. Блочная и групповая релаксации — некоторое подмножество (блок) координат x_i придают одинаковые или различные (групповая релаксация) приращения таким образом, чтобы свести к нулю одну из невязок или их сумму.

3. Сверхрелаксация — изменяют знак или величину некоторых невязок так, чтобы улучшить последующую минимизацию невязок.

Основное преимущество итерационных методов связано с их малой чувствительностью к погрешностям округления и случайным ошибкам в сходящемся процессе вычислений, приводящим к увеличению затрат времени, но не влияющим на конечный результат.

Рассмотренными выше методами можно вычислить и обратную матрицу $F = A^{-1}$, для чего нужно n раз решить систему уравнений $AX = Q$, заменяя каждый раз вектор Q очередным j -м столбцом единичной матрицы. В этом случае в соответствии с формулой (80а) каждое решение даст значения элементов j -го столбца обратной матрицы.

Решение системы с большим числом уравнений можно свести к решению системы с меньшим числом уравнений. Для этого система из n уравнений разбивается на две подсистемы (блоки) с $a < n$ и $b = n - a$ уравнениями

$$\begin{aligned} A_{aa}X_a + A_{ab}X_b &= Q_a; \\ A_{ba}X_a + A_{bb}X_b &= Q_b. \end{aligned} \quad (89a)$$

Исключая вектор X_b из первого блока, получают

$$\begin{aligned} (A_{aa} - A_{ab}A_{bb}^{-1}A_{ba})X_a &= Q_a - A_{ab}A_{bb}^{-1}Q_b; \\ A_{ba}X_a + A_{bb}X_b &= Q_b. \end{aligned} \quad (89b)$$

Решая первый блок относительно вектора X_a , подставляют полученные значения во второй блок для вычисления вектора X_b .

Последовательно применяя блочные преобразования, можно свести решение системы уравнений к решению блоков с небольшим числом уравнений. В частности, если исключается одна переменная x_n , то элементы матрицы $A_{aa}^1 = (A_{aa} - A_{ab}A_{bb}^{-1}A_{ba})$ первого блока вычисляют по формулам

$$a_{ij}^1 = a_{ij} - \frac{a_{in}a_{nj}}{a_{nn}} = \frac{1}{a_{nn}} \begin{vmatrix} a_{ij} & a_{in} \\ a_{nj} & a_{nn} \end{vmatrix}. \quad (90)$$

Блочные преобразования особенно удобны в тех случаях, когда интерес представляют составляющие только вектора X_a . В частности,

если первый блок системы (89а) содержит только два уравнения, а во втором блоке $Q_b = 0$, то решение исходной системы уравнений с $n > 2$ переменными сводится к решению системы из двух уравнений

$$\frac{1}{\Delta_{11,22}} \begin{bmatrix} \Delta_{22} & -\Delta_{21} \\ -\Delta_{12} & \Delta_{11} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad (91)$$

где $\Delta_{11,22}$ — кратный минор, равный определителю исходной матрицы A n -го порядка с вычеркнутыми первыми и вторыми строками и столбцами; Δ_{ij} — алгебраические дополнения матрицы A .

Разбиение системы уравнений на блоки (89а) соответствует разбиению матрицы A на подматрицы (клетки). Операции над элементами клеточных и обыкновенных матриц аналогичны, что позволяет понизить размеры матриц, над которыми непосредственно выполняются операции. Например, для матриц A и B с одинаковым разбиением на клетки

$$A + B = \begin{bmatrix} A_{11} + B_{11} & A_{12} + B_{12} \\ A_{21} + B_{21} & A_{22} + B_{22} \end{bmatrix};$$

$$AB = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{bmatrix}. \quad (92)$$

Обращение матрицы A порядка $n = p + q$, разбитой на 4 клетки с диагональными квадратными неособенными клетками порядка p и q , сводится к вычислению элементов клеток обратной матрицы $F = A^{-1}$ по формулам

$$F_{11} = (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1};$$

$$F_{12} = -F_{11}A_{12}A_{22}^{-1}; \quad (93a)$$

$$F_{21} = -A_{22}^{-1}A_{21}F_{11};$$

$$F_{22} = A_{22}^{-1} - A_{22}^{-1}A_{21}F_{11}.$$

Если клетка A_{22} содержит только один элемент a_{nn} исходной матрицы A , то элементы клеток обратной матрицы

$$F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1}$$

можно вычислить по формулам

$$1/f_{nn} = a_{nn} - A_{n1}A_{11}^{-1}A_{1n};$$

$$F_{1n} = -A_{11}^{-1}A_{1n}/f_{nn};$$

$$F_{n1} = -f_{nn}A_{n1}A_{11}^{-1}; \quad (93b)$$

$$F_{11} = A_{11}^{-1} + f_{nn}A_{11}A_{1n}A_{n1}A_{11}^{-1}.$$

Обращая аналогично подматрицы низшего порядка клетки A_{11} , можно понизить до желательного порядка непосредственно обращаемую подматрицу.

Решение системы уравнений с разреженной (имеющей нулевые элементы) матрицей A можно свести к операциям над матрицами низшего порядка при расширении исходной системы уравнений окаймлением исходной матрицы. Исходная матрица A разбивается на клетки с диагональными неособенными клетками A_1, \dots, A_k так, чтобы вне диагональных клеток содержалось минимальное число ненулевых элементов. Тогда каждую строку (столбец) таких элементов можно перенести в окаймляющие строку или столбец с диагональным элементом $e_{n+1, n+1} = 1$ так, чтобы при исключении окаймления согласно формуле (90) восстанавливалась исходная матрица. В этом случае получается расширенная система уравнений, записываемая в блочном виде как

$$\begin{bmatrix} A_1 & 0 & \cdots & 0 & B_1 \\ 0 & A_2 & \cdots & 0 & B_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & A_k & B_k \\ C_1 & C_2 & \cdots & C_k & E \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_k \\ X_0 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ \cdots \\ Q_k \\ 0 \end{bmatrix}, \quad (94a)$$

где E — единичная матрица; A_i — квадратные неособенные подматрицы исходной матрицы A ; X_0 — дополнительный вектор неизвестных переменных, при исключении которого расширенная система (94a) преобразуется в исходную. Решение исходной системы уравнений по клеткам расширенной системы (94a) определяется формулами [17]:

$$Z = - \sum_{i=1}^k C_i A_i^{-1} B_i; \quad K = E + Z; \quad X_0 = K^{-1} Z; \\ X_i = A_i^{-1} (Q_i - B_i X_0); \quad i = 1, 2, \dots, k. \quad (94b)$$

Буквенные (символические) решения систем уравнений (79) сводятся к разложению числителей и знаменателей функций f_{ji} в формулах (80) по символам элементов матрицы A или их составляющих и свободных членов q_i . Алгебраические дополнения $\Delta_{ab} = (-1)^{a+b} M_{ab}$ только знаком могут отличаться от миноров M_{ab} , равных определителям матрицы A с вычеркнутыми a -й строкой и b -м столбцом. Следовательно, задача буквенного решения системы уравнений (79) сводится к разложению определителей порядка k на многочлены вида

$$\Delta = \sum_{q=1}^s (-1)^{\delta_q} \prod_{r=1}^k (a_{ij})_r; \quad i, j \in \{1, 2, \dots, n\}, \quad (95)$$

где s — число ненулевых членов определителя; δ_q — декремент подстановки q -го члена; $(a_{ij})_r$ — элемент a_{ij} матрицы A (или его составляющая), являющийся r -м сомножителем q -го члена. При

в каждом q -м члене все индексы i и все индексы j сомно-
жителей различны.

Классический способ разложения определителя по строкам или столбцам достаточно громоздок. Более удобны алгоритмы метода обобщенных чисел [23], основанные на описании структуры матрицы A частично-упорядоченным множеством (матричным числом β_M) индексов m и l ее ненулевых элементов и построении изоморфного определителю множества подстановок (контурного числа β) согласно формуле

$$\Delta = \det \beta = \det (\beta_M)_{\text{mod } 2}. \quad (96)$$

Формулу (96) можно реализовать при помощи следующего алгоритма.

1. По заданной матрице A составляется матричное число β_M из n строк, каждая m -я строка которого образована множеством номеров l столбцов, на пересечении которых находятся ненулевые элементы m -й строки матрицы.

2. Элементы l первой строки матричного числа записываются с достаточно большими пробелами, под каждым из них проводится линия, под которой выписываются все элементы l второй строки матричного числа, отличающиеся от отделенного линией элемента первой строки.

3. Аналогичные операции повторяются над записанными элементами второй и последующих строк до разложения всех строк матричного числа. «Висячие» выписанные элементы, под которыми нет элементов следующей строки, а также линии между одиночными элементами вычеркиваются.

4. В полученном контурном числе столбцы (образованные элементами нижней и расположенными над ними элементами верхних строк, включая отделенные линиями общие элементы столбцов) с нечетным числом инверсий (числом больших номеров l , расположенных над каждым элементом l столбца) подчеркиваются.

5. Элементы l каждого столбца вместе с номерами m соответствующих строк заменяются произведениями изоморфных элементов a_{ml} матрицы. Произведениям, изоморфным подчеркнутым столбцам, приписывается отрицательный знак.

Миноры M_{ab} (и соответственно кратные миноры) раскладываясь аналогично по матричному числу с вычеркнутой a -й строкой и элементами $l = b$ остальных строк.

Пример 10. Разложить по символам элементов определитель Δ и алгебраическое дополнение Δ_{25} матрицы

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & a_{15} \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ 0 & 0 & a_{33} & 0 & a_{35} \\ 0 & a_{42} & a_{43} & a_{44} & 0 \\ -a_{51} & 0 & 0 & a_{54} & a_{55} \end{bmatrix}.$$

В соответствии с приведенным алгоритмом находим

$$\Delta = \det \beta = \det \begin{bmatrix} 1 & 2 & 5 \\ 2 & 1 & 2 & 3 \\ 3 & 3 & 5 \\ 4 & 2 & 3 & 4 \\ 5 & 1 & 4 & 5 \end{bmatrix} \pmod{2} = \det \begin{bmatrix} 1 & 2 & 5 \\ 2 & \frac{1}{2} & 3 \\ 3 & \frac{3}{5} & 5 \\ 4 & \frac{4}{3} & \frac{4}{2} & 4 \\ 5 & \frac{5}{4} & 4 \end{bmatrix} =$$

$$= \det \begin{bmatrix} 1 & 2 & 5 \\ 2 & \frac{2}{3} & \frac{1}{3} & \frac{1}{2} \\ 3 & \frac{3}{5} & \frac{5}{3} & \frac{5}{3} & 3 \\ 4 & \frac{4}{3} & \frac{2}{4} & \frac{2}{4} \\ 5 & \frac{5}{4} & \frac{4}{5} & \frac{4}{1} & \frac{4}{1} \end{bmatrix} = a_{11}[a_{22}(a_{33}a_{44}a_{55} + a_{35}a_{43}a_{54}) - a_{23}a_{35}a_{42}a_{54}] - a_{12}[a_{23}a_{35}a_{44}a_{51} + a_{21}(a_{33}a_{44}a_{55} + a_{35}a_{43}a_{54})] - a_{15}(a_{22}a_{33}a_{44}a_{51} + a_{21}a_{33}a_{42}a_{54});$$

$$\Delta_{25} = -M_{25} = -\det \begin{bmatrix} 1 & 2 \\ 3 & 3 \\ 4 & 2 & 3 & 4 \\ 5 & 1 & 4 \end{bmatrix} \pmod{2} = -\det \begin{bmatrix} 1 & 2 \\ 3 & 3 & 3 \\ 4 & \frac{2}{4} & 4 \\ 5 & 4 & 1 \end{bmatrix} =$$

$$= a_{11}a_{33}a_{42}a_{54} + a_{12}a_{33}a_{44}a_{51}.$$

В тех случаях, когда в буквенном виде необходимо представить лишь некоторые из элементов матрицы a_1, \dots, a_m или их составляющих, при $m \leq n$ используется разложение

$$\Delta = \Delta^0 + \sum_{r=1}^m a_r \Delta_r^0 + \sum_{r=1}^m \sum_{s=r+1}^m a_r a_s \Delta_{rs}^0 + \dots + a_1 \dots a_m \Delta_{1 \dots m}^0, \quad (97a)$$

где определитель Δ^0 и кратные алгебраические дополнения $\Delta_r^0 \dots$ вычисляют по матрице A при $a_1 = \dots = a_m = 0$. При $m > n$ последний член разложения (97a) заменяется суммой

$$\sum_{r_1=1}^m \dots \sum_{r_n=r_{n-1}+1}^m a_{r_1} \dots a_{r_n} \Delta_{r_1 \dots r_n}^0.$$

Если описываемый буквами элемент a_r является составляющей элементов матрицы A , расположенных в строках a и c и столбцах b и d , то в разложении (97a) каждый индекс r алгебраических дополнений заменяется индексом $(a+c)(b+d)$. Суммарное алгебраическое дополнение [22]

$$\Delta_{(a+c)(b+d)} = \Delta_{ab} - \Delta_{ad} - \Delta_{cb} + \Delta_{cd} \quad (97b)$$

может только знаком отличаться от минора $M_{(a+c)(b+d)} = (-1)^{a+b} \Delta_{(a+c)(b+d)}$, равного определителю матрицы A , в которой вычеркнуты a -я строка и b -й столбец, а их элементы добавлены к соответствующим элементам c -й строки и d -го столбца.

Подставляя вместо буквенных символов элементов их численные значения, с помощью разложений (97) можно найти численное решение системы линейных уравнений (79). Однако такой способ практически пригоден лишь для матриц A низкого порядка или имеющих большое число нулевых элементов.

Исключение составляют часто встречающиеся в инженерной практике задачи моделирования системой линейных уравнений (79) физических объектов с изменяющимися переменными q_i воздействий или параметров a_{ij} активных каналов причинно-следственной связи. В этих случаях изменяющиеся величины обычно отображают табличными моделями и для различных сочетаний дискретных значений этих переменных в заданных интервалах их изменения вычисляют множества значений x_i , характеризующих их зависимость от изменяющихся элементов модели.

Многократное численное решение системы уравнений для каждого набора дискретных значений ее изменяющихся элементов (многовариантный анализ) требует значительных затрат времени. При большом числе расчетных точек более эффективно составление буквенными методами расчетных формул, в которых согласно разложения вида (97а) буквами отображены только изменяющиеся величины. Для составления таких формул могут потребоваться относительно большие затраты времени, поэтому обычно представляют интерес не все, а лишь некоторые из искомых переменных x_i . Вычисление этих переменных для каждого набора значений изменяющихся элементов модели по расчетным формулам выполняется относительно быстро.

Кроме того, при плохо обусловленных задачах (близость определителя матрицы A к нулю, ее асимметрия или большое различие в значениях элементов) операционная погрешность при использовании точных методов численного решения может оказаться недопустимо большой. Ее также можно уменьшить при использовании буквенных методов с устранением подобных членов при разложении определителей.

Системы нелинейных уравнений формально представимы в операторной форме (8), внешне совпадающей с формой записи линейных уравнений, но для решения таких систем обычно используют методы математического программирования.

7. МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Во многих инженерных задачах приходится учитывать не только ограничения на численные значения элементов математической модели, описывающей свойства физического объекта системой равенств и неравенств, но и дополнительные требования качества (оптимальности) результата. Если математическая модель содержит систему уравнений ранга r , то при числе неизвестных $n = r$ их значения определяются решением этой системы. Однако при $n > r$ единственный способ определения всех n неизвестных заключается в подборе $n - r$ переменных для удовлетворения всех условий задачи, включая требования оптимальности.

Целенаправленный поиск (оптимизация) результата решения в последнем случае достигается методами математического програм-

мирования*, основанными на составлении вещественной целевой функции искомых переменных, принимающей экстремальное значение при наилучшем (оптимальном) удовлетворении условий задачи, и целенаправленном поиске экстремумов этой функции.

Основная задача математического программирования заключается в отыскании оптимальных значений переменных $x_i; i = 1, 2, \dots, n$, соответствующих экстремуму целевой функции

$$\Phi(X) = \Phi(x_1, \dots, x_n) \quad (98a)$$

при ограничениях в виде системы равенств

$$H_j(X) = 0; j = 1, 2, \dots, k \quad (98b)$$

и системы неравенств

$$G_j(X) \geq 0; j = k + 1, k + 2, \dots, m. \quad (98v)$$

Двусторонние ограничения переменных вида $\alpha_i \leq x_i \leq \beta_i$ сводятся к односторонним заменой их на $\beta_i - x_i \geq 0; x_i - \alpha_i \geq 0$, а системы неравенств (98v) могут заменяться системой равенств $g_j = G_j(X); j = \overline{1, m}$ с дополнительными неотрицательными переменными g_j .

Всякому набору значений оптимизируемых переменных x_i в n -мерном пространстве соответствует вектор $X = [x_1, \dots, x_n]^T$, направленный из начала координат в точку X , координаты которой равны значениям переменных и соответствуют некоторому значению целевой функции $\Phi(X)$. Точки этого пространства, в которых целевая функция имеет одинаковые значения, называют поверхностями равного уровня, а точки $\Phi(X)$, в которых целевая функция экстремальна, точками минимума или максимума целевой функции или оптимума изменяемых переменных. При точном решении задачи оптимизации точки оптимума лежат на пересечении поверхностей (98b) внутри или на границе допустимой области, ограниченной поверхностями (98v). Если целевая функция многоэкстремальна, то минимумы (максимумы) называют локальными, а наименьший (наибольший) из них — глобальным. Минимум $\Phi(X)$ равен максимуму $-\Phi(X)$.

В зависимости от вида функций (98) различают линейное и нелинейное программирование. Основная задача линейного программирования заключается в вычислении переменных x_i , удовлетворяющих системе линейных уравнений

$$g_j = \sum_{i=1}^n (-a_{ij}x_i) + a_j \geq 0; \quad j = \overline{1, m} \quad (99a)$$

при максимуме целевой функции

$$\Phi = \sum_{i=1}^n b_i x_i. \quad (99b)$$

* Математическое программирование (как раздел математики) непосредственно не связано с программированием как процессом составления программ автоматических вычислений для ЭВМ.

Эта задача обычно решается симплекс-методом, основанным на вычислительном процессе, включающем отыскание опорного решения, удовлетворяющего системе уравнений (99а) с последующим переходом к оптимальному решению, соответствующему максимуму функции (99б).

На первом этапе симплекс-метода по заданной системе уравнений (99а) составляют таблицу вида

	$-x_1$	$-x_2$...	$-x_n$	1
g_1	a_{11}	a_{12}	...	a_{1n}	a_1
...
g_n	a_{n1}	a_{n2}	...	a_{nn}	a_n
g_{n+1}	$a_{n+1, 1}$	$a_{n+1, 2}$...	$a_{n+1, n}$	a_{n+1}
...
g_m	a_{m1}	a_{m2}	...	a_{mn}	a_m
Φ	$-b_1$	$-b_2$...	$-b_n$	0

не включающую ограничений на знак переменных $x_i \geq 0$ или $x_i \leq 0$. Эту таблицу с помощью модифицированных жордановых исключений $r \leq n$ переменных, где r — ранг прямоугольной матрицы коэффициентов a_{ij} , приводят к симплекс-таблице, принимающей при $r = n$ вид

	$-g_1$	$-g_2$...	$-g_n$	1
g_{n+1}	$\alpha_{n+1, 1}$	$\alpha_{n+1, 2}$...	$\alpha_{n+1, n}$	α_{n+1}
...
g_m	$\alpha_{m, 1}$	$\alpha_{m, 2}$...	α_{mn}	α_m
Φ	β_1	β_2	...	β_n	B

с новыми значениями коэффициентов.

Уравнения для исключенных переменных

$$x_i = -(\alpha_{i1}g_1 + \alpha_{i2}g_2 + \dots + \alpha_{in}g_n) + \alpha_i; \quad i = \overline{1, n}$$

в симплекс-таблицу не записывают, хранят для определения переменных x_i после оптимизации переменных g_j .

Каждое k -е модифицированное жорданово исключение выполняется по следующим правилам:

выбирают разрешающий элемент $\alpha_{pq}^{(k-1)}$ исходной таблицы;

вычисляют новые значения коэффициентов таблицы по формулам

$$\alpha_{ji}^{(k)} = \alpha_{ji}^{(k-1)} - \frac{\alpha_{pi}^{(k-1)} \alpha_{jq}^{(k-1)}}{\alpha_{pq}^{(k-1)}}; \quad \alpha_{pi}^{(k)} = -\frac{\alpha_{pi}^{(k-1)}}{\alpha_{pq}^{(k-1)}}; \quad \alpha_{jq}^{(k)} = \frac{\alpha_{jq}^{(k-1)}}{\alpha_{pq}^{(k-1)}};$$

$$\alpha_{pq}^{(k)} = \frac{1}{\alpha_{pq}^{(k-1)}}; \quad (100b)$$

переменные g_p и g_q в таблице переставляют. В случае 0-уравнения, когда переменная $g_p = 0$, после исключения по ведущему элементу α_{pq} столбец таблицы с номером q вычеркивают.

В результате исключения переменных x_i в таблице (100a) их меняют местами с переменными g_i и после устранения всех $r \leq n$ свободных переменных получается соответствующая таблице (100b) задача меньшей размерности: оптимизировать переменные g_j , удовлетворяющие системе уравнений

$$g_j = \sum_{i=1}^n (-\alpha_{jig_i}) + \alpha_j; \quad j = n+1, \dots, m \quad (101a)$$

при максимальном значении функции

$$\Phi = \sum_{i=1}^n \beta_i + B. \quad (101b)$$

Система уравнений (100a) соответствует опорному решению, если все свободные члены α_j уравнений неотрицательны. В противном случае в симплекс-таблице выбирают $\alpha_s = \alpha_{s, n+1} < 0$ и в s -й строке отрицательный элемент α_{st} . После этого в t -столбце выбирают разрешающий элемент α_{ht} , для которого отношение α_h/α_{ht} минимально, и производят жорданово исключение по этому элементу.

Подобный процесс повторяют до положительности всех свободных членов, соответствующей опорному решению. Если в некоторой строке свободный член $\alpha_s = 0$ и имеются неотрицательные элементы α_{si} , то из таблицы вычеркивают эту строку и столбцы с $\alpha_{si} \geq 0$. Если положительны все элементы строки с отрицательным свободным членом, то система уравнений (101a) несовместна и задача оптимизации неразрешима.

При неотрицательности всех коэффициентов $\beta_i \geq 0$ опорное решение оптимально и $\max \Phi = B$ соответствует $g_i = 0; i = 1, \dots, n$ и $g_j = \alpha_j; j = n+1, \dots, m$. В противном случае выбирают коэффициент $\beta_t < 0$ с наибольшим модулем, в t -м столбце выбирают разрешающий положительный элемент с минимальным α_h/α_{ht} для h -й строки и производят исключение по этому элементу. Повторяя подобные исключения, приходят к оптимальному решению ($\beta_i \geq 0$) или, при отсутствии положительных элементов в очередном t -м столбце, к выводу о неразрешимости задачи.

В процессе оптимизации может произойти зацикливание, когда после нескольких исключений приходят к тому же разрешающему элементу. В этом случае необходимо выбрать другой разрешающий элемент. При очередном исключении целесообразно вначале вычислять свободные члены и элементы β_i последней строки, так как в случае их неотрицательности решение оптимально и вычисление остальных коэффициентов таблицы не имеет смысла.

Задача нелинейного программирования возникает, когда все или некоторые функции (98) нелинейны или полилинейны (имеются произведения переменных). Нелинейное итерационное программирование при детерминированном поиске на каждой k -й итерации сводится к отысканию для точки $X^{(k-1)}$ в n -мерном пространстве переменных x_i точки

$$X^{(k)} = X^{(k-1)} + S^{(k)} = X^{(k-1)} + \alpha \xi^{(k)} \quad (102)$$

с минимальным значением $\Phi(X^{(k)}) < \Phi(X^{(k-1)})$. При этом по информации, полученной на предыдущих итерациях, выбирают вектор предпочтительного направления $\xi^{(k)} = S^{(k)} / \|S^{(k)}\|$ и отыскивают оптимальную длину вектора $\alpha = \|S^{(k)}\|$, соответствующую ближайшей точке минимума $\Phi(\alpha)$. Таким образом, задача минимизации целевой функции на каждой итерации разбивается на выбор предпочтительного направления ξ и одномерный поиск минимума $\Phi(\alpha)$ вдоль этого направления,

Одномерный детерминированный поиск минимума целевой функции $\Phi(\alpha)$ сводится к отысканию точки, в которой производные $\Phi'(\alpha)$ и $\Phi''(\alpha)$ равны нулю. Однако отыскание этой точки может оказаться сложным и обычно такой поиск разбивается на выделение интервала нахождения минимума и его последующее сокращение до выполнения условия

$$|\Phi(\alpha_k) - \Phi(\alpha_{k-1})| \leq \varepsilon_1 \text{ или } |\alpha_k - \alpha_{k-1}| \leq \varepsilon_2, \quad (103)$$

где числа ε_1 и ε_2 определяют требуемую или возможную точность вычисления $\Phi(\alpha)$ и α .

Выделяют интервал минимума обычно с переменным шагом, вычисляя $\Phi(\alpha)$ в точках $\alpha_k = \alpha_{k-1} + hf(k)$, $k = 1, 2, \dots$ с начальным шагом h и увеличивая его в соответствии с функцией $f(k)$ номера шага, например, $f(k) = a^k$. В этом случае при $a = 2$ получают формулу прогрессии, а при $a = \tau \approx 1,618\dots$ (число золотого сечения)

$$\Delta \alpha_k = \tau \Delta \alpha_{k-1}; \quad \alpha_k = \alpha_{k-1} + \tau (\alpha_{k-1} - \alpha_{k-2}).$$

Если уменьшающееся в точках α'_{p-2} , α_{p-1} значение функции возрастает в точке α_p согласно неравенству

$$\Phi(\alpha_{p-2}) > \Phi(\alpha_{p-1}) < \Phi(\alpha_p),$$

то искомый интервал (α_{p-2}, α_p) найден.

Среди различных методов сокращения интервала аппроксимацией целевой функции или делением интервала к наиболее эффективным относится метод золотого сечения. Он заключается в последовательном делении интервала на отрезки с отношением длин $\tau \approx 1,618\dots$. При использовании этого метода по нижней α_n и верхней α_b границам интервала вычисляют две точки

$$\underline{\alpha}_k = \frac{\tau \alpha_{n,k} + \alpha_{b,k}}{1 + \tau}; \quad \bar{\alpha}_k = \frac{\tau \alpha_{b,k} + \alpha_{n,k}}{1 + \tau} \quad (104a)$$

внутри интервала и значение функции в этих точках. Сравнивая значения функции, выбирают одну из границ сокращенного интервала:

$$\text{если } \Phi(\underline{\alpha}_k) \leq \Phi(\bar{\alpha}_k), \text{ то } \alpha_{n,k+1} = \underline{\alpha}_k; \quad \alpha_{b,k+1} = \bar{\alpha}_k; \quad (104b)$$

$$\text{если } \Phi(\underline{\alpha}_k) > \Phi(\bar{\alpha}_k), \text{ то } \alpha_{n,k+1} = \bar{\alpha}_k; \quad \alpha_{b,k+1} = \underline{\alpha}_k,$$

и снова производят вычисления по формулам (104a) для новой длины интервала. Если вычислять только одну точку внутри интервала, то вследствие ошибок округления возможна потеря малого интервала, содержащего точку минимума.

В зависимости от порядка производных, используемых при выборе предпочтительных направлений, различают методы нелинейного программирования нулевого, первого и высших порядков. Применение этих методов [28] связано с громоздкими вычислениями, объем которых резко возрастает при увеличении количества оптимизируемых переменных.

В методах нулевого порядка предпочтительное направление выбирают по критериям, не связанным с вычислением производных. К таким методам, в частности, относится метод координатного спуска, при котором одномерный поиск ведется по направлениям, параллельным координатным осям, выбираемым определенным образом.

Координатный спуск, при котором выбор очередной координаты одномерного поиска определяется наибольшим отрицательным значением производной целевой функции по координатам, относится к методам первого порядка. Среди градиентных методов, связанных с вычислением производных, широко используется метод скорейшего спуска, в котором предпочтительное направление выбирается противоположным направлению градиента в многомерном пространстве переменных.

Методы нелинейного программирования (в частности, методы координатного и скорейшего спусков) используются также для решения систем линейных и, особенно, нелинейных уравнений. Многие из рассмотренных методов решения таких уравнений также относятся по существу к методам нелинейного программирования, так как основаны на минимизации разностей между приближенными и точными значениями искомых переменных.

Большинство методов математического программирования предназначены для решения задач безусловной оптимизации, соответствующей отсутствию ограничений на значения оптимизируемых переменных. Задачи с ограничениями можно свести к задачам безусловной оптимизации, но в этом случае сложность решения, как правило, существенно возрастает. Вместе с тем решение многих практических задач оптимизации упрощается при надлежащем выборе их математической формулировки.

Следует отметить, что ординаты экстремумов функции одной переменной обычно определяют как корни уравнения, полученные приравниванием нулю производной функции по ее аргументу. Для функций многих переменных такой метод, как правило, непригоден, так как полученное уравнение оказывается сложнее исходного, и поэтому целесообразно использовать численные методы математического программирования. Во многих случаях это относится и к функции одной переменной

Пример 11. Минимизировать функцию

$$y = \sqrt{100 + x^2} + \sqrt{25 + (20 - x)^2}$$

в области $x > 0$ с точностью определения y до 6 верных знаков.

Дифференцирование исходного выражения приводит к еще более сложному нелинейному уравнению. Поэтому будем искать минимум заданной функции методом половинного деления. Так как эта функция монотонна при $x < 0$ и при $x > 20$, то в качестве исходного примем интервал нахождения минимума $0 < x < 20$. Результаты одномерного поиска методом половинного деления приведены в табл. 8.

Таблица 8

i	x_i	y_i	i	x_i	y_i
0	0	30,61552	8	13,28125	25,00008
1	20	27,36067	9	13,35932	25,00002
2	10	25,32247	10	13,32031	25,00000
3	15	25,09882	11	13,30078	25,00003
4	12,5	25,02168	12	13,31054	25,00001
5	13,75	25,00574	13	13,33008	25,00000
6	13,125	25,00135	14	13,32519	25,00000
7	13,4375	25,00035			

Проверка погрешностей результата последней итерации с помощью вычислительного графа рассмотренным ранее методом показывает, что заданная точность достигнута.

Задачи оптимизации являются наиболее сложными, но и наиболее значительными задачами инженерного проектирования, цель которого заключается в построении математических моделей физических объектов, удовлетворяющих заданным критериям качества (оптимальности).

Задача приближенного решения системы уравнений общего вида (линейных или нелинейных)

$$\delta_i \equiv f_i(x_1, x_2, \dots, x_n) = 0; i = 1, 2, \dots, n \quad (105)$$

с n неизвестными эквивалентна задаче минимизации некоторой функции невязок δ_i , например,

$$\Phi(x_1, x_2, \dots, x_n) = \sum_{i=1}^n [f_i(x_1, x_2, \dots, x_n)]^2. \quad (106)$$

При решении этой задачи приближенными методами выбирают некоторые начальные приближения $x_i^{(0)}$ и последовательно находят приближения (102), сходящиеся к корням уравнений (105) при определенных условиях, зависящих от вида уравнений и выбранного метода решения. Различные итерационные методы отличаются способом выбора в n -мерном пространстве переменных x_i направления $\xi^{(k)}$ и длины шага $\alpha^{(k)}$, приводящих на $(k+1)$ -м шаге к минимуму функции $\Phi(\alpha)$, аппроксимируемой рядом Тейлора или, при табличном задании, интерполяционным многочленом.

Системы нелинейных уравнений (105) с вещественными корнями обычно решают итерационным методом Ньютона или его различными модификациями. Рассматривая множества аргументов x_i и функций f_i как векторы $X = [x_1 \dots x_n]^t$ и $F = [f_1 \dots f_n]^t$, уравнения (105) можно записать в матрично-векторной форме как $F(X) = 0$. Если найдено k -е приближение одного из векторов корней $X^{(k)} = [x_1^{(k)} \dots x_n^{(k)}]^t$, то $X = X^{(k)} + E^{(k)}$, где $E = [\varepsilon_1^{(k)} \dots \varepsilon_n^{(k)}]^t$ — вектор, описывающий погрешности корня. Для непрерывно дифференцируемой в области решения функции

$$F(X^{(k)} + E^{(k)}) = F(X^{(k)}) + F'(X^{(k)})E^{(k)} = 0,$$

где матрицу

$$F'(X) = W = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

называют матрицей Якоби.

При неособенности (отличии от нуля определителя) этой матрицы $E^{(k)} = -W^{-1}F(X^{(k)})$ и можно построить процедуру последовательных приближений

$$X^{(k+1)} = X^{(k)} - W_{(k)}^{-1}F(X^{(k)}); \quad k = 0, 1, 2, \dots, \quad (107)$$

где в качестве нулевого приближения принимают грубое приближение одного из корней.

Вычисления упрощаются при использовании модифицированного метода Ньютона, описываемого формулой

$$X^{(k+1)} = X^{(k)} - W_{(0)}^{-1}F(X^{(k)}); \quad k = 0, 1, 2, \dots, \quad (108)$$

где обратная матрица Якоби вычисляется один раз для начального приближения.

Систему нелинейных уравнений (105), приведенную к виду

$$x_i = \varphi(x_1, x_2, \dots, x_n); \quad i = 1, 2, \dots, n \quad (109)$$

или, в векторной форме, $X = \Phi(X)$, можно решить и с помощью различных других итерационных методов, в частности, методом последовательных приближений (простых итераций) согласно формуле

$$X^{(k+1)} = \Phi(X^{(k)}); \quad k = 0, 1, 2, \dots, \quad (110)$$

сходящейся при определенных условиях [8] к искомым корням уравнения. Для решения систем нелинейных (иногда и линейных) уравнений используют и различные другие методы нелинейного программирования — в частности, методы координатного и скользящего спуска.

ВЫЧИСЛЕНИЯ НА НЕПРОГРАММИРУЕМЫХ МИКРОКАЛЬКУЛЯТОРАХ

1. ВХОДНЫЕ ЯЗЫКИ НЕПРОГРАММИРУЕМЫХ МИКРОКАЛЬКУЛЯТОРОВ

Эффективность вычислений при помощи электронных микрокалькуляторов, как и ЭВМ других классов, определяется затратами времени на получение результата вычислений с требуемой точностью. Эта эффективность максимальна при полном использовании всех возможностей микрокалькулятора и рациональной организации вычислительного процесса, включая выбор метода и алгоритма вычислений, а также формы хранения промежуточных результатов и методики контроля их точности. Свойства калькулятора как вычислительного средства определяются зависящими от его конструкции способами представления, ввода, вывода и хранения чисел, а также правилами выполнения операций над этими числами.

Вывод результата вычислений и контроль вводимых чисел практически во всех электронных микрокалькуляторах обеспечивается высвечиванием десятичного представления числа на индикаторе. В микрокалькуляторах различных типов числа представимы с фиксированной или плавающей запятой. При фиксированной запятой и k десятичных разрядах для высвечиваемых на индикаторе чисел диапазон их представления ограничен нижним пределом $A_{\min} = \pm 10^{-k+1}$ (если, как обычно, первый разряд индикатора используется для высвечивания нуля перед запятой) и верхним пределом $A_{\max} = \pm (10^k - 1)$. Например, при $k = 8$ интервал представления чисел ограничен пределами $A_{\min} = 0,0000001$ и $A_{\max} = 99999999$.

Интервал представления чисел существенно расширяется при плавающей запятой и лежит в пределах от $A_{\min} = \pm 10^{-k+1} \times 10^{-(10^l-1)}$ до $A_{\max} = \pm (10^k - 1) \cdot 10^{10^l-1}$, где l — число разрядов порядка числа. Следовательно, при $k = 8$ и $l = 2$ максимальная ширина интервала представления чисел ограничена пределами $A_{\min} = \pm 0,0000001 \cdot 10^{-99}$ и $A_{\max} = \pm 99999999 \cdot 10^{99}$. В микрокалькуляторах некоторых типов этот интервал может быть не-

сколько уже предельного, а операции выполняются над числами, десятичные представления которых имеют на 1—2 младших разряда больше, чем высвечиваемые на индикаторе.

Результаты вычислений с плавающей запятой (при отличии порядка числа от нуля) высвечиваются на индикаторах микрокалькуляторов в стандартной форме с расположением запятой после первой значащей цифры в отличие от обычных ЭВМ, где запятая в маинтиссе нормализованного представления числа расположена перед первой значащей цифрой.

Числа, меньшие по модулю нижнего предела представления числа A_{\min} , относятся к области «машинного нуля», а большие по модулю верхнего предела A_{\max} — к области «машинной бесконечности». При попадании результата вычислений на микрокалькуляторе в область машинной бесконечности на индикаторе высвечивается условный знак, указывающий на переполнение устройств хранения чисел в калькуляторе.

Операции над числами выполняются микрокалькулятором по командам, подаваемым пользователем посредством нажатия соответствующих клавишей. Множество символов выполняемых операций (операторов), указываемых на клавиатуре калькулятора, и правила подачи команд нажатием клавишей образуют входной язык (язык управления заданиями) микрокалькулятора данного типа. Этот входной язык, как и любой, обычный или машинный язык, характеризуется лексикой, грамматикой и семантикой.

В обычных языках различают устную и письменную формы. Аналогично входные языки микрокалькуляторов кроме письменной формы для записи последовательности команд и, при необходимости, «ответов» микрокалькулятора, высвечиваемых на индикаторе, имеют активную форму, заключающуюся в самом процессе «разговора» с микрокалькулятором посредством нажатия клавишей и регистрацией высвечиваемых «ответов». В дальнейшем при отсутствии специальных оговорок под входным языком микрокалькулятора будем подразумевать его письменную форму.

Словами входного языка микрокалькулятора являются символы команд (операторы), а множество операторов, указанных на клавиатуре, образуют словарный запас или лексикон входного языка микрокалькулятора данного типа. Различные операторы можно отнести к одной из трех основных групп — ввода (представления) чисел, автоматических вычислений и служебных.

Первая группа включает операторы ввода в микрокалькулятор цифр от 0 до 9, десятичного разделительного знака (запятой или точки), изменения знака числа /—/ (обычно на калькуляторе высвечивается только отрицательный знак) и ввода порядка числа в микрокалькуляторах с плавающей запятой.

Вторая группа обязательно содержит операторы двухместных (производимых над двумя operandами) арифметических операций и, кроме того, может содержать операторы автоматического вычисления одноместных и двухместных элементарных функций.

Третья группа содержит операторы стирания (сброса) чисел, хранимых в устройствах внешней памяти (при выключении микрокалькулятора содержимое всех устройств внешней памяти также стирается), операторы обмена содержимым устройств памяти, префиксные операторы, обеспечивающие использование одних и тех же клавиш для подачи различных команд и т. д.

Во входных языках микрокалькуляторов различных типов одинаковые операторы иногда обозначают различными символами, а одинаковыми символами — различные операторы. Поэтому существенное значение имеет семантика входного языка, определяющая соответствие между символикой и назначением операторов.

Перечень правил, определяющих последовательность ввода необходимых для вычислений по выбранному алгоритму операторов (команд), образует грамматику входного языка микрокалькулятора. Множество команд, записанное в виде последовательности символов вводимых операторов (слов), образует предложение на входном языке. При нарушении грамматических правил такое предложение, как и в обычных языках, не имеет смысла и приводит к ошибочным результатам. Предложение, написанное на машинном языке, обычно называют программой. Однако для определенности программой будем называть только предложение, вводимое в память программируемого микрокалькулятора для последующих автоматических вычислений по этой программе.

Микрокалькуляторы различных типов отличаются запоминающими устройствами для хранения операндов, результатов операций и программ автоматических вычислений. Эти устройства разделяются на внешние, управляемые нажатием соответствующих клавиш, и внутренние, в которых постоянно хранятся программы, реализующие операторы входного языка. В дальнейшем будем рассматривать только внешние запоминающие устройства.

Запоминающее устройство для хранения одного числа называется регистром памяти. В простейших микрокалькуляторах имеется только два операционных регистра памяти — индикаторный, в котором хранится высвечиваемый на индикаторе operand. и регистр, в который заносится второй operand при двухместных операциях. Микрокалькуляторы некоторых типов имеют несколько операционных регистров, в частности — дополнительный регистр памяти, в котором хранятся числа, используемые как постоянные коэффициенты (константы) при вычислениях, или результаты промежуточных вычислений, переносимые из других операционных регистров.

В некоторых микрокалькуляторах предусмотрены стековые устройства памяти, состоящие из нескольких регистров, содержимое которых сдвигается при вводе operandов в стек или выводе из него. При этом различают обычные стеки, из которых числа выводятся в обратной последовательности (первым выводится последнее введенное число), стеки-очереди, в которых ввод и вывод производятся с различных концов, и кольцевые стеки (деки), в которых как

ввод, так и вывод чисел может производиться с любого конца. Наиболее удобны запоминающие устройства с произвольной выборкой (ЗУПВ), имеющие несколько регистров памяти с независимым вводом и выводом содержимого каждого регистра.

В зависимости от назначения и входного языка микрокалькуляторы можно условно разбить на три основных класса — арифметические, инженерные и программируемые.

Арифметические микрокалькуляторы обеспечивают автоматическое выполнение только арифметических операций — сложения, вычитания, умножения и деления. Для сравнения рассмотрим два типа арифметических микрокалькуляторов — «Электроника Б3-04» с экономичным питанием от одного элемента напряжением 1,5 В и «Электроника Б3-24» с универсальным питанием.

Лексикон входного языка микрокалькулятора типа «Электроника Б3-04» содержит (рис. 7, а) операторы ввода цифр и десятичного разделительного знака (точки), арифметические операторы, символы которых ($+=$, $-=$, \times , \div) частично совмещены со знаком вывода результата, оператор К занесения в память константы и оператор С стирания содержимого индикаторного регистра. При нажатии клавиши К на месте крайнего правого (знакового) разряда индикатора высвечивается буква К. При переполнении содержимое индикаторного регистра стирается.

Грамматические правила составления предложений на входном языке аналогичны обычным правилам ручной записи арифметических операций в строку, за исключением использования скобок. Для отрицательных значений операндов символ $-=$ вводится после них (а не перед ними, что характерно для знака «—» в обычной записи). Например, для вычисления $17 - 9 + 6$ вводится предложение*

$$|C|17|+|=|9|-=|6|+=|(14)|, \text{ а не } |C|17|-=|9|+|=|6|+=|(2)|,$$

* В записи предложений на входных языках микрокалькуляторов символы операторов и operandов разделяются пробелами или вертикальными линиями; в круглых скобках при необходимости обозначаются высвечиваемые результаты операций. Многозначные operandы в общем случае можно записывать слитно (например, |3, 06| вместо |3|, |0| |6|).

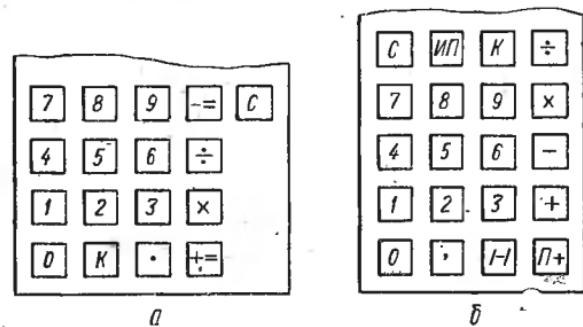


Рис. 7. Клавиатура арифметических микрокалькуляторов типа:

а — «Электроника Б3-04»; б — «Электроника Б3-24»

что приводит к ошибочному результату. Предусмотрена возможность деления и умножения чисел на одно и то же число (константу). Например, для умножения числа 20 на 4, 5 и 6 составляется предложение

$$|C|K|20|\times|4|+ = |(80)|5|+ = |(100)|6|+ = |(120)|.$$

Если при этом перед вводом каждого множителя нажимать клавишу оператора \times , то очередной результат будет умножаться на вводимый операнд

$$|C|K|20|\times|4|+ = |(80)|\times|5|+ = |(400)|\times|6|+ = |(2400)|.$$

Если же в таком предложении опустить оператор K, то очередной результат будет суммироваться с вводимым операндом:

$$|C|20|\times|4|+ = |(80)|5|+ = |(85)|6|+ = |(91)|.$$

Деление с константой производится вводом предложений вида

$$|C|K|20|\div|4|+ = |(5)|6|+ = |(3,333333)|.$$

Входной язык микрокалькулятора типа «Электроника Б3-24» (рис. 7, б) подобен рассмотренному, причем операторы + и — (аналогично операторам += и ==) выполняют также функцию вывода результата. Однако язык этого микрокалькулятора имеет три дополнительных оператора — оператор /—/ изменения знака содержимого индикаторного регистра, оператор П+ суммирования содержимого индикаторного регистра с содержимым дополнительного регистра памяти и оператор ИП переноса содержимого дополнительного регистра в индикаторный. Наличие дополнительного регистра значительно облегчает выполнение последовательности разнородных арифметических операций без записи на бумаге промежуточных результатов. Например, вычисления по формуле $(1,2 \times 4 + 8 : 4) \times 3$ выполняются вводом предложения

$$|C|1,2|\times|4|\Pi+|8|\div|4|+|ИП|\times|3|+|(20,4)|.$$

Инженерные микрокалькуляторы, кроме обязательных арифметических операторов, иногда имеют операторы, выполняющие функции скобок при обычной записи последовательности операций в строку. Для входных языков инженерных микрокалькуляторов характерны операторы автоматического вычисления элементарных функций. К одноместным элементарным функциям относят степенные x^a (например, $1/x$, x^2 , \sqrt{x} и т. д.), показательную (экспоненциальную) e^x , тригонометрические и обратные тригонометрические, логарифмические $\ln x$ и $\lg x$ и т. п. Примером двухместной элементарной функции аргументов x и y может служить функция x^y .

Входные языки инженерных микрокалькуляторов могут существенно отличаться как лексикой, так и грамматическими правилами. Для сравнения рассмотрим инженерные микрокалькуляторы типов «Электроника Б3-14М» и «Электроника Б3-18А».

Простейший инженерный микрокалькулятор типа «Электроника Б3-14М» отличается удобным входным языком, содержащим (рис. 8, а) кроме операторов ввода десятичных знаков и арифметических операторов, также операторы вычисления обратной функции $1/x$ и извлечения квадратного корня $\sqrt{\cdot}$. Кроме того, имеются операторы вывода (высвечивания) результата $=$, стирания последнего введенного операнда СК и стирания содержимого всех регистров памяти С. Правила ввода операторов и операндов в основном совпадают с обычными правилами записи в строку, что упрощает пользование микрокалькулятором. Например, вычисление $17 \times 3 : 4 - 6$ реализуется предложением

$$|C| 17 | \times | 3 | \div | 4 | - | 6 | = |(6,75)|.$$

Результаты вычисления обратной величины и извлечения корня высчитываются на индикаторе без нажатия клавиши $=$ (например, $|C| 36 | \sqrt{ } | (6) |$). После получения на индикаторе результата умножения при дополнительном нажатии клавиши $=$ происходит умножение этого результата на предыдущий множитель. Поэтому для возведения в целую степень n числа x достаточно вычислить произведение x и x и дополнительно $n - 1$ раз нажать клавишу $=$. Например, для вычисления $5,1^5$ вводится предложение

$$|C| 5,1 | \times | 5,1 | = | = | = | = |(3450, 252)|.$$

Если возводимое в степень число многоразрядно, то для упрощения вычислений целесообразно выбрать первым множителем единицу, умножить ее на аргумент и после этого n раз нажать клавишу $=$, например,

$$|C| 1 | \times | 5,1 | = | = | = | = |(3450,252)|.$$

При вводе в индикаторный регистр рассматриваемого микрокалькулятора десятичной дроби оператор ввода нуля можно опускать, например, для ввода числа 0.532 достаточно ввести предложение $|, | 5 | 3 | 2 |$.

Значительно большими возможностями автоматического (по внутренним программам) вычисления элементарных функций отличается входной язык инженерного микрокалькулятора типа «Электроника Б3-18А». Десятичное представление чисел на

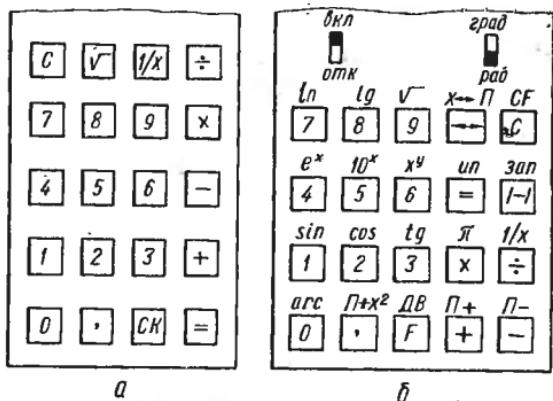


Рис. 8. Клавиатура инженерных микрокалькуляторов типа:
а — «Электроника Б3-14М»; б — «Электроника Б3-18М»

индикаторе этого микрокалькулятора восьмиразрядное с фиксированной запятой, причем запятая не занимает отдельного разряда. В связи с этим диапазон представления чисел лежит в пределах от $A_{\min} = \pm 0,0000001$ до $A_{\max} = \pm 99999999$. Если результат операции превышает A_{\max} по модулю, то в крайнем левом (знаковом) разряде индикатора высвечивается знак переполнения (точка). Отрицательный знак высвечивается в том же разряде при вводе оператора $/-$ после ввода числа. Ввод дробного числа с нулевой целой частью допустимо начинать с десятичного разделительного знака (запятой).

Операторы входного языка, символы которых обозначены на клавишиах (рис. 8, б), вводятся нажатием этих клавиш. При вводе арифметических операторов ($+$, $-$, \times и \div) выполняется предыдущая операция, а операция, соответствующая вводимому оператору, выполняется после ввода операнда и следующего арифметического оператора или оператора $=$. При дополнительных вводах оператора $=$ последний операнд становится константой, с которой выполняется последняя арифметическая операция, например,

$$|64| \div |8| = |(8)| = |(1)| = |(0,125)| \dots ; \quad |1| \times |2,18| = \\ |(2,18)| = |(4,7524)| = |(10,360232)| \dots \text{ или } |64| \div |8| = |(8)| 2 = \\ |(0,25)| 4 = (0,5) \dots .$$

Для ввода операторов, символы которых указаны над клавищами микрокалькулятора рассматриваемого типа, необходимо предварительно нажать клавишу префиксного оператора F. Результаты большинства совмещенных (требующих ввода префиксного оператора) операций высвечиваются без ввода оператора $=$, например,

$$F|\pi|(3,1415926); |0,5|F| \quad 1/x|(2); \quad |0,5|F| \ln|(-0,693147); \\ |0,5|F| \lg|(-0,30103); |0,5|F| \sqrt| (0,7071067); |0,5|F| e^x(1,648721); \\ |0,5|F| 10^x|(3,162277); \quad |0,5|F| \sin| (0,479425); \quad |0,5|F| \sin \\ (0,008726); |0,5|F| \arctan| \tg| (26,56505^\circ) \text{ и т. д. Однако для вы-} \\ \text{числения функции } x^y \text{ необходимо ввести предложение вида} \\ |x|F|x^y|y| = |(|), \text{ содержащее оператор } =, \text{ например, } |10|F| \times \\ |x^y|(2,302585)|2| = |(99,99995).$$

При вычислениях числа могут храниться в трех регистрах — двух операционных (индикаторном и рабочем) и дополнительном регистре памяти с независимой выборкой. Оператор \leftrightarrow предназначен для обмена содержимых индикаторного и рабочего регистров, а оператор $|F|x \leftrightarrow \Pi|$ — индикаторного и регистра памяти. При вводе $|F|ИП|$ на индикаторе высвечивается содержимое регистра памяти, а при вводе $|F|ЗАП|$ содержимое индикаторного регистра записывается также в регистр памяти. При вводе предложений $|x|F|\Pi +|$ или $|x|F|\Pi -|$ число x складывается с содержимым регистра памяти или вычитается из него, а при вводе предложения $|x|F|\Pi + x^2|$ содержимому регистра памяти добавляется квадрат x .

Для очистки операционных регистров при режиме совмещенной функции или переполнении трижды нажимается клавиша С; первое нажатие снимает режим совмещенной функции или переполнение, второе очищает индикаторный и третье — рабочий регистры. Регистр памяти очищается вводом предложения |С|F|ЗАП|0|. Оператор ДВ («данные восстановлены») предназначен для исправления ошибочно введенной цифры. Например, если во вводимом числе 2,356 последняя цифра должна быть исправлена на 7, то вводится предложение |F|ДВ|(2,35)| С|7| (2,357). Для восстановления результата предыдущей операции после первой цифры следующего операнда вводится предложение |F|ДВ|.

При вычислении функций $1/x$, \sqrt{x} и логарифмов рабочего регистра не изменяется и можно выполнять смешанные (цепные) операции. После вычисления остальных совмещенных функций рабочий регистр автоматически очищается.

В инженерных микрокалькуляторах некоторых типов имеется большее число операционных регистров, что обеспечивает выполнение цепных операций. Примером может служить микрокалькулятор типа «Электроника Б3-19М». Диапазон представления чисел в этом микрокалькуляторе ограничен пределами от $A_{\min} = \pm 1 \cdot 10^{-99}$ до $A_{\max} = \pm 9,999999 \cdot 10^{99}$ при плавающей запятой и семи разрядах для мантиссы (при округлении отбрасыванием). Входной язык (рис. 9) по лексике близок к языку микрокалькулятора типа «Электроника Б3-18А», но отличается рядом грамматических правил. В частности, при выполнении двухместных операций их операторы вводятся после операндов (первым может быть результат предыдущей операции), разделенных оператором ↑ засылки первого операнда в операционный регистр.

Микрокалькулятор рассматриваемого типа имеет регистр памяти m с произвольной выборкой и операционный стек с тремя регистрами x , y и z . Оператор ↑ засыпает содержимое индикаторного регистра x в регистр y , а содержимое последнего — в регистр z . Ввод двухместного оператора после операнда заносит результат операции в регистр x , а прежнее содержимое регистра z переносит в регистр y . Ввод двухместного оператора после двухместного оператора заносит результат операции над содержимым регистров x и y в регистр x , не изменяя прежнего содержимого регистров y и z . Ввод операнда после двухместного оператора сдвигает содержимое регистра x в регистр y , а регистра y — в регистр z .

Описанные свойства операционного стека позволяют выполнять цепные операции «со скобками» без дополнительной записи проме-

e^x	x^y
C _x	π
sin	1-1
7	F
arc sin	cos
4	tan
√	x ← m
lg	arc cos
1	arc tan
C	m ← x
0	x
	1/x
	ln
	-
	x ← y
	CF
	↑
	+

Рис. 9. Клавиатура инженерного микрокалькулятора «Электроника Б3-19М»

жуточных результатов на бумаге. Например, вычисления по формуле $(3 \times 4) + (6 : 2) \times 5 - (8 : 4)$ выполняются с помощью одного предложения. Если под каждым оператором такого предложения выписать содержимое операционных регистров x , y и z , то получим следующую запись, наглядно характеризующую связи между регистрами операционного стека:

C	3	↑ 4	×	6	↑ 2	÷ +	↑ 5	×	8	↑ 4	÷ -							
$x =$	0	3	3	4	12	6	6	2	3	15	15	5	75	8	8	4	2	73
$y =$	0	0	3	3	0	12	6	6	12	12	15	15	12	75	8	8	75	75
$z =$	0	0	0	0	0	12	12	12	12	12	12	12	12	75	75	75	75	

Так как входные языки непрограммируемых микрокалькуляторов достаточно близки, то в дальнейшем при отсутствии специальных оговорок будем использовать стандартизованную письменную форму входного языка непрограммируемых микрокалькуляторов со следующими правилами:

1. Предложение начинается оператором С очистки операционных регистров и оканчивается оператором = вывода результатов.
2. Одноместные операторы записываются после операнда, двухместные — между соответствующими операндами.
3. Высвечиваемые на индикаторе числа при необходимости записываются в круглых скобках.
4. Перенос содержимого индикаторного регистра в N -й регистр памяти или N -ю графу вычислительного бланка отображается оператором PN, а вызов этого числа в индикаторный регистр — оператором FN.
5. Операторы отделяются пробелами или вертикальными линиями, кроме операторов ввода одного операнда, записываемых (при отсутствии оговорок) слитно.

Например, вычисление функции $y = 1,21 \ln 5 + \sqrt{1 - \sin 0,3}$ с указанием высвечиваемых результатов на этом стандартизированном языке можно описать предложением

|C|5|ln|(1,609436)|×|12,1||(19,47418) P1|0,3|sin|(0,295520)|/-|+
+|1|(0,704479)|V|(0,839333)|+|F1|=|(20,31351).

Грамматика стандартизованного языка достаточно проста, а составленные на нем предложения легко переводятся на входные языки микрокалькуляторов конкретных типов с учетом их семантики и лексики.

2. ОСОБЕННОСТИ ВЫЧИСЛЕНИЙ НА МИКРОКАЛЬКУЛЯТОРАХ

Основное отличие вычислений на электронных микрокалькуляторах от ручных вычислений и от вычислений при помощи универсальных ЭВМ связано с ограниченным диапазоном представления

чисел при фиксированном количестве разрядов (обычно 7–10 разрядов в микрокалькуляторах различных типов для представления числа в естественной форме или мантиссы числа в стандартной форме и 2 разряда для порядка чисел).

При ручных вычислениях с помощью карандаша и бумаги количество разрядов десятичных представлений чисел принципиально не ограничено, что позволяет находить результат с любой точностью, изменяя при необходимости разрядность результата каждой операции. Большинство ЭВМ работает при фиксированной разрядности operandов, но при необходимости ее можно увеличить (как правило, в кратное число раз) для контроля или повышения точности вычислений.

Фиксированная разрядность operandов в микрокалькуляторах обычно обеспечивает требуемую точность инженерных вычислений при небольшом числе операций. Однако при увеличении числа операций погрешность округления накапливается, что может существенно снизить точность окончательного результата. Более того, результат вычислений может оказаться полностью ошибочным вследствие попадания промежуточных результатов в область машинного нуля или бесконечности, хотя действительное значение окончательного результата лежит в диапазоне представления чисел микрокалькулятора используемого типа. При этом особенно опасно попадание промежуточных результатов в область машинного нуля, так как при машинной бесконечности на индикаторе высвечивается условный знак переполнения (а в микрокалькуляторах некоторых типов вычисления автоматически прекращаются), тогда как высвечивание на индикаторе нуля может не обратить внимания пользователя на возможность принципиальной ошибки в конечном результате вычисления.

Невозможность проверки точности результата вычислений на микрокалькуляторе увеличением разрядности operandов заставляет обращать серьезное внимание на оценку погрешностей при выполнении сложных вычислений. При оценке погрешностей рассмотренными в гл. 1 методами необходимо также учитывать особенности микрокалькулятора используемого типа — способ представления чисел, разрядность operandов и способ автоматического округления результатов операций.

В большинстве микрокалькуляторов результат выполнения арифметической операции округляется до рабочей разрядности k простым отбрасыванием младших избыточных разрядов. В этом случае абсолютная погрешность округления ограничена единицей младшего разряда высвечиваемого результата и, следовательно, зависит от его порядка n .

Для микрокалькуляторов со стандартной формой представления чисел с плавающей запятой абсолютная погрешность округления не превышает

$$\varepsilon_a \leqslant 10^{n-k+1},$$

а относительная погрешность

$$\varepsilon_o \leqslant 10^{n-k+1}/a \approx 10^{1-k}/a_m,$$

где a_m — первая значащая цифра высвечиваемого числа a .

При естественной форме представления чисел приведенная оценка абсолютной погрешности округления справедлива лишь при $n \geqslant 0$. В противном случае абсолютную погрешность необходимо оценивать как

$$\varepsilon_a \leqslant 10^{1-k}.$$

Относительная погрешность округления при естественной форме представления чисел для $n \geqslant 0$ оценивается по приведенной выше формуле, а для $n < 0$ — по формуле

$$\varepsilon_o \leqslant 10^{1-k}/a \approx 10^{1-k-n}/a_m.$$

Следует учитывать, что ошибка округления может оказаться недопустимо большой, если не принять соответствующих мер (например, нормирования операндов предварительным приведением их к стандартной форме). Так, при возведении в квадрат числа 0,0004 на 8-разрядном микрокалькуляторе типа «Электроника Б3-18М» результат 0,0000001 с порядком $n = -7$ по приведенной выше оценке вычислен с относительной погрешностью $\varepsilon_o \leqslant 10^{1-8+7}/1 = 1$ или 100% (при истинном значении абсолютной погрешности 0,00000006 или относительной погрешности 60%).

Если в микрокалькуляторе используемого типа результат операции округляется симметрично, то все приведенные оценки следует уменьшать вдвое. Однако для результатов автоматического вычисления по внутренним программам элементарных функций погрешность может превышать единицу последнего разряда, на что обычно указывается в инструкциях по эксплуатации микрокалькуляторов соответствующих типов.

Ограниченнная разрядность операндов в микрокалькуляторах приводит к нарушению законов коммутативности, ассоциативности и дистрибутивности, справедливых для арифметических операций над представлениями чисел с неограниченной разрядностью.

Закон коммутативности (результат умножения не зависит от перестановки сомножителей) может нарушаться при умножении (делении) нескольких операндов вследствие округления частных произведений. Например, при вычислениях на 8-разрядном микрокалькуляторе с представлением чисел в естественной форме (с фиксированной запятой) при различной последовательности умножений получим

$$C \ 0,0015 \times 0,01328 \times 0,02 \times 10,54 \times 100,5 = (0,0003115);$$

$$C \ 0,0015 \times 0,01328 \times 10,54 \times 0,02 \times 100,5 = (0,0004120);$$

$$C \ 0,0015 \times 100,5 \times 0,01328 \times 10,54 \times 0,02 = (0,0004220).$$

В первом случае произведение первых трех сомножителей округляется до 7 значащих цифр после запятой и отбрасываемые значащие цифры не участвуют в последующих операциях, что приводит к операционной погрешности окончательного результата $\varepsilon_o \approx 25\%$.

Результат умножения нескольких операндов с фиксированной запятой имеет наименьшую погрешность при соблюдении следующего правила: при каждой операции умножается самый большой из самый малый из оставшихся сомножителей (включая результат предыдущего умножения). При этом может оказаться необходимым хранить некоторые из результатов умножений, что неудобно при отсутствии в микрокалькуляторе дополнительных регистров памяти. Поэтому обычно используют более простое правило: после первого умножения самого большого на самый малый сомножитель очередной результат y_i умножают на самый малый, если $y_i > 1$, или самый большой, если $y_i < 1$, из оставшихся сомножителей.

Этим правилом следует руководствоваться и в тех случаях умножения (деления) нескольких операндов, когда промежуточный результат может оказаться в области машинного нуля ($y_i < A_{\min}$) или машинной бесконечности ($y_i > A_{\max}$). Например, при вычислениях на 8-разрядном микрокалькуляторе с представлением чисел в естественной форме

$$\begin{aligned} C 0,00001 \times 0,001 \times 0,1 \times 10 \times 1000 \times 10000 &= (0); \\ C 10000 \times 1000 \times 10 \times 0,1 \times 0,001 \times 0,00001 &= (\infty); \\ C 0,00001 \times 10000 \times 0,001 \times 1000 \times 0,1 \times 10 &= (1), \end{aligned}$$

и, следовательно, правильный результат получается лишь в последнем случае.

При вычислениях на микрокалькуляторе со стандартной формой представления чисел операционная погрешность умножения нескольких операндов в значительно меньшей степени зависит от последовательности вычисления частных произведений, так как округляются лишь мантиссы чисел:

$$\begin{aligned} C 1,19 \cdot 10^5 \times 1,239 \cdot 10^{-1} \times 1,3459 \cdot 10^4 \times 1,45679 &= \\ = (2,8908663 \cdot 10^8); \quad C 1,45679 \times 1,19 \cdot 10^5 \times 1,3459 \cdot 10^4 \times 1,239 \cdot 10^5 &= \\ = (2,8908662 \cdot 10^8); \quad C 1,45679 \times 1,3459 \cdot 10^4 \times 1,239 \cdot 10^1 \times 1,19 \cdot 10^5 &= \\ = (2,8908661 \cdot 10^8). \end{aligned}$$

Однако и в этом случае при большом числе операций погрешности округлений накапливаются и погрешность окончательного результата может оказаться недопустимо большой.

Для предотвращения получения ошибочного результата вследствие попадания результата промежуточных вычислений в область машинных нуля или бесконечности при умножении нескольких операндов в стандартной форме также следует использовать приведенное выше правило. Например, при вычислениях на микрокалькуля-

торе с восемью разрядами мантиссы и двумя разрядами порядка операндов получим

$$\begin{aligned} C 2 \cdot 10^{-96} \times 0,0001 \times 3 \cdot 10^{95} \times 140000 &= (0); \\ C 3 \cdot 10^{95} \times 140000 \times 2 \cdot 10^{-96} \times 0,0001 &= (\infty); \\ C 2 \cdot 10^{-96} \times 3 \cdot 10^{95} \times 140000 \times 0,0001 &= (8,4). \end{aligned}$$

Так как операционная погрешность умножения чисел с естественной формой представления значительно больше, чем при стандартной форме, то при использовании микрокалькуляторов с фиксированной запятой может оказаться целесообразным предварительно записать сомножители в стандартной форме, найти произведение их мантисс и сумму порядков и записать окончательный результат. Например, нормировав сомножители к стандартной форме $1,5 \cdot 10^{-3}$; $1,328 \times 10^{-2}$; $2 \cdot 10^{-2}$; $1,054 \cdot 10^1$; $1,005 \cdot 10^2$, вычислив произведение мантисс

$$C 1,5 \times 1,328 \times 2 \times 1,054 \times 1,005 = (4,2201316)$$

и сложив порядки сомножителей, получим результат $4,2201316 \times 10^{-4}$ с семью верными значащими цифрами вместо трех при непосредственном умножении операндов с фиксированной запятой.

При сложении (вычитании) двух чисел в калькуляторе любого типа слагаемые уравниваются по порядку и округляются младшие разряды. При суммировании нескольких операндов может произойти нарушение закона ассоциативности («результат сложения не изменяется при перестановке слагаемых»). Например, при вычислениях на 8-разрядном калькуляторе с округлением отбрасыванием

$$\begin{aligned} C 10000,305 + 1,0095 + 0,00195 &= (10001,334); \\ C 1,0095 + 0,00195 + 10000,305 &= (10001,335). \end{aligned}$$

Операционная ошибка в подобных случаях накапливается при увеличении числа слагаемых. Поэтому при суммировании большого числа слагаемых на микрокалькуляторе любого типа следует придерживаться следующего правила: слагаемые разбиваются на группы чисел, близких по порядку, суммируются слагаемые каждой из групп, после чего складываются частные суммы, начиная с наименьших. Если используется микрокалькулятор с естественной формой представления чисел, то может оказаться целесообразной предварительная запись слагаемых, близких по порядку, в стандартной форме.

Резкое снижение точности результата может произойти, если близкие по модулю слагаемые имеют различные знаки, так как относительная погрешность разности двух чисел $\varepsilon_o = \varepsilon_a / |a - b|$ тем больше, чем меньше их разность. Поэтому при вычитании близких чисел следует попытаться так преобразовать формулу вычислений, чтобы исключить вычитание. Например, если d — малое число, то целесообразно использовать преобразования

$$\begin{aligned} (a + d)^2 - a^2 &= 2ad + d^2; \quad \sqrt{a+d} - \sqrt{a} = \\ &= d / (\sqrt{a+d} + \sqrt{a}) \text{ и т. п.} \end{aligned}$$

Необходимо учитывать возможность нарушения закона дистрибутивности при смешанных операциях с округлением промежуточных результатов. Так, если вычитание близких чисел исключить не удается, то следует начинать вычисления с таких вычитаний, так как погрешность вычисления $(a - b)c$ меньше погрешности вычисления $ac - bc$. Более того, если числа a и b имеют общий сомножитель, то его целесообразно вынести за скобки перед вычитанием. Например, при вычислении на 8-разрядном микрокалькуляторе с естественным представлением чисел результат вычисления $(6,250001 \times 16 - 25,000003 \times 4) \cdot 100000$ будет равен нулю, тогда как при вынесении за скобки сомножителя 4 получим $(6,250001 \times 4 + 25,000003) \cdot 400000 = 4$.

Точность результата сложения можно существенно повысить, уравнивая разрядность слагаемых по младшим разрядам и разбивая разряды (начиная с младших) на группы с $k - 1$ разрядами, если число слагаемых не более 10, или с меньшим числом разрядов при большем числе слагаемых. Суммируя содержимое разрядов в каждой группе, можно получить результат сложения со всеми значащими цифрами. Например, уравнивая по младшим разрядам числа 0,00195; 1,00950; 10 000,30500 при вычислениях на 8-разрядном калькуляторе, находим сумму чисел в их семи последних разрядах 01,31645 и сумму 100 содержащего трех левых разрядов и получаем результат 10001,31645 со всеми верными цифрами. Подобный прием целесообразно использовать при вычитании близких чисел.

Следует также учитывать, что если в операционном регистре микрокалькулятора с k разрядами мантиссы записано число порядка $n_1 \leq k$, то любое другое число, большее машинного нуля, но имеющее порядок $n_2 \leq k - n_1$, при сложении будет нулем относительно первого числа. Например, для 8-разрядного микрокалькулятора $356400 + 0,009 = (358400)$ при представлении чисел в микрокалькуляторе как в естественной, так и в стандартной форме.

При вычислениях на инженерных микрокалькуляторах источником погрешности может быть ограниченная разрядность результата вычисления элементарных функций. В этих случаях целесообразно использовать нормирование операндов или изменить алгоритм вычислений. Например, при $x > \ln A_{\max}$ вычисление e^x приведет к переполнению, но если разбить x на несколько слагаемых, меньших $\ln A_{\max}$, вычислить экспоненциальные функции от каждого слагаемого, произведение мантисс и сумму порядков значений этих функций, то результат будет получен с k значащими цифрами. Вычисления по формуле

$$y = (e^{30x} - e^{29x})/(e^{29x} - 1)$$

при $x = 10$ приведут к переполнению даже при использовании микрокалькулятора с $A_{\max} = 9,9999999 \cdot 10^{99}$, но простое преобразование

$$y = (e^x - 1)/(1 - e^{-29x}) = 22025,46$$

даст результат с точностью до семи знаков.

Влияние округления операндов иногда удается свести к минимуму при использовании итерационных процедур. Так, для определения $y = \sqrt{x}$ с точностью до 1 верных знаков на 8-разрядном микрокалькуляторе вычисления можно организовать следующим образом. Представив $y = y_0 + \Delta y$, где y_0 — такое приближение искомого корня в стандартной форме, точный квадрат которого имеет не более восьми знаков, получим $y_0^2 + 2\Delta y y_0 + \Delta y^2 = x$. Учитывая малость Δy , можно принять $\Delta y_1 \approx (x - y_0^2)/2y_0$ и последующие значения поправок вычислять по рекуррентной формуле

$$\Delta y_i = [(x - y_0^2) - \Delta y_{i-1}^2]/2y_0,$$

где значения Δy_i записываются в стандартной форме, что легко достигается умножением правой части на нормирующий множитель 10^q с соответствующим выбором порядка q . Итерационный процесс сходится, если выполняется неравенство $(x - y_0^2)/y_0 < 1$.

Например, для вычисления $y = \sqrt{2}$ по этому алгоритму

$$y_0 = 1,414; \quad x - y_0^2 = 2 - 1,999396 = 0,000604;$$

$$\Delta y_1 = \frac{0,000604}{2 \cdot 1,414} = \frac{6,04}{2,828} \cdot 10^{-4} = 2,135785 \cdot 10^{-4};$$

$$\Delta y_2 = \frac{6,04 - 4,5615775 \cdot 10^{-4}}{2,828} \cdot 10^{-4} = 2,1356237 \cdot 10^{-4};$$

$$\Delta y_3 = \frac{6,04 - 4,5608885 \cdot 10^{-4}}{2,828} \cdot 10^{-4} = 2,1356237 \cdot 10^{-4}.$$

Складывая «ручную» полученную поправку с y_0 и округляя результат до требуемой разрядности, получим $y = 1,4142135624$. Для проверки результата запишем $y^2 = (y_0 + \Delta y)^2 = 1,414^2 + 2 \times 1,414 \times 2,1356237 \cdot 10^{-4} + 2,1356237^2 \cdot 10^{-8}$. Вычисливmantиссы слагаемых, после «ручного» сложения последних получим $y = 1,99999999998885$ с 10 верными знаками после запятой, так как уже второе слагаемое имеет не большую точность.

Описанная итерационная процедура особенно удобна при вычислении выражений вида $y = a - \sqrt{a^2 - b}$ с $b \ll a^2$, встречающихся при решении квадратных уравнений. В этом случае итерационное соотношение упрощается;

$$y_i = (b - y_{i-1}^2)/2a \text{ при } y_0 = b/2a.$$

Ошибка вычислений при условии $y_i = y_{i-1}$ не превышает единицы младшего разряда результата. При использовании подобной процедуры для вычисления π (см. пример 6) ошибка округления при каждом i -м удвоении сторон вписанного многоугольника уменьшается до

$$\Delta a_i/a_i \leqslant 10^{-7} + \Delta a_{i-1}/a_{i-1} \approx i \cdot 10^{-7}.$$

Минимальная погрешность не более $15 \cdot 10^{-7}$ достигается в этом случае при $i = 11$ (практически $\pi_{11} = 3,1415918$ с погрешностью около $8 \cdot 10^{-8}$).

Выбор алгоритма, обеспечивающего требуемую точность вычислений при минимальных затратах времени, связан с особенностями входного языка микрокалькулятора используемого типа. Практически все инженерные микрокалькуляторы обеспечивают выполнение цепных вычислений без дополнительной записи промежуточных результатов. Поэтому целесообразно так преобразовать заданную расчетную формулу к удобному для цепных вычислений виду, чтобы свести к минимуму количество вводов операндов и регистраций промежуточных результатов. Например, несложное преобразование

$$y = a \left(x_1 + \frac{x_2}{12} + \frac{x_3}{36} \right) = \left(\left(\frac{x_3}{3} + x_2 \right) : 12 + x_1 \right) a$$

позволяет произвести все необходимые операции без записи промежуточных результатов.

Для удобства вычислений следует описать расчетную формулу предложением входного языка микрокалькулятора используемого типа. Например, приведенная выше формула на стандартном языке описывается предложением $|C|F3|\div|3|+|F2|\div|12|+|F1|\times|Fa|=|$, где символами FN обозначены операторы ввода переменных x_3 , x_2 , x_1 и a соответственно.

После выбора алгоритма вычислений целесообразно (а часто необходимо) нормировать переменные и коэффициенты расчетных выражений. Нормирование заключается в замене искомого результата и переменных их линейными функциями и позволяет уравнивать по порядку входящие в расчетные соотношения коэффициенты. Например, при вычислении на 8-разрядном калькуляторе с естественной формой представления чисел функции

$$y = 2 \cdot 10^{-12}x^3 + 6,2 \cdot 10^{-8}x^2 + 3,8 \cdot 10^{-4}x + 6$$

результат может иметь чрезмерную погрешность, так как первые два коэффициента многочлена будут машинными нулями для этого микрокалькулятора, в то время как их произведение на степень аргумента может оказаться больше остальных членов. Так, при $x = 10^6$ значение $y = 2664$, тогда как без учета первых двух членов $y = 44$. Простая замена $z = x \cdot 10^{-4}$ (или $x = z10^4$) приводит рассматриваемый многочлен к удобному для вычислений нормированному виду $y = 2z^3 + 6,2z^2 + 3,8z + 6$.

При вычислении корней степенного многочлена его удобно предварительно нормировать делением всех коэффициентов на один из них (обычно на коэффициент при высшей степени аргумента). Значение дробно-рациональной функции не изменится при нормировании ее коэффициентов делением на значение одного из них.

Часто основной интерес представляет не абсолютное значение анализируемой функции $y(x)$, а характер ее изменения в интервале

аргумента при исследовании физических зависимостей, численном решении уравнений, поиске экстремумов функций и т. п. В этих случаях целесообразно нормировать не только аргумент, но и значение функции, сокращая тем самым число операций и вводимых коэффициентов. Так, дифференциальное уравнение

$$\frac{d^2x}{dt^2} - 300(1 - 25x^2) \frac{dx}{dt} + 10^{10}x = 0$$

заменами $\tau = t \cdot 10^5$ и $z = 5x$ после подстановки

$$\frac{dx}{dt} = \frac{d(z/5)}{d\tau} \frac{d\tau}{dt} = \frac{10^5}{5} \frac{dz}{d\tau}; \quad \frac{d^2x}{dt^2} = \frac{d}{d\tau} \left(\frac{dx}{dt} \right) \frac{d\tau}{dt} = \frac{10^{10}}{5} \frac{d^2z}{d\tau^2}$$

и деления на $10^{10}/5$ приводится к уравнению

$$\frac{d^2z}{d\tau^2} - 0,003(1 - z^2) \frac{dz}{d\tau} + z = 0$$

с одним отличным от единицы коэффициентом. Окончив вычисления, несложно выполнить денормирование всех переменных. Даже в тех случаях, когда нормирование для уменьшения погрешностей не обязательно, оно желательно для упрощения ввода коэффициентов и переменных за счет сокращения их разрядности и уменьшения числа операций.

При составлении по выбранному алгоритму предложений на входном языке используемого микрокалькулятора необходимо четко представлять себе содержимое его регистров памяти после каждой операции. Это позволит составить наиболее короткую последовательность команд с минимальным количеством вводимых операндов и регистраций промежуточных результатов.

В качестве примера опишем вычисление площади треугольника со сторонами a , b и c и полупериметром $p = (a + b + c)/2$ по формуле Герона

$$S = \sqrt{p(p - a)(p - b)(p - c)}$$

на входном языке микрокалькулятора типа «Электроника Б3-18М»

$$|C|a|3AP| + |b| + |c| \div |2| = |-|X \leftrightarrow \Pi| + |\leftrightarrow| \times |= \\ |X \leftrightarrow \Pi| - |b| + |\leftrightarrow| \times |X \leftrightarrow \Pi| = |X \leftrightarrow \Pi| - |c| \times |IP| = |V|.$$

Для вычисления S по этому предложению достаточно пять вводов операндов (один раз a и дважды b и c) без записи промежуточных результатов. Содержание операционных регистров x и y и регистра памяти Π при вводе первых 10 операндов и операторов изменяется следующим образом:

$ C a 3AP \div b + c \div 2 = V \dots$
$x = 0 \quad a \quad a \quad a \quad b \quad a+b \quad c \quad a+b+c \quad 2 \quad p \quad \dots$
$y = 0 \quad 0 \quad 0 \quad a \quad a \quad b \quad a+b \quad c \quad a+b+c \quad 2 \quad \dots$
$\Pi = 0 \quad a \quad \dots$

Эта же задача решается при помощи микрокалькулятора типа «Электроника Б3-19М» вводом предложения, содержащего меньшее число операторов (при том же числе вводов операндов)

|C|a|↑|b|↑|c|+|+|2|÷|x→m|X↔Y|—|m→x|×|↑|↑|
|m→x|b|—|×|↑|↑|m→x|c|—|x|V|.

Это достигается использованием обмена содержимым регистров памяти, которое на первых 10 шагах изменяется следующим образом:

		C		a		↑	b		↑	c		+		+		2		÷	x→m	...
x =	0	a	a	b	b	c	b+c	a+b+c	2	p	p	...								
y =	0	0	a	a	b	b	a	a	a+b+c	a	a	...								
z =	0	0	0	0	a	a	a	a	a	a	a	...								
m =	0	0	0	0	0	0	0	0	0	0	0	...								

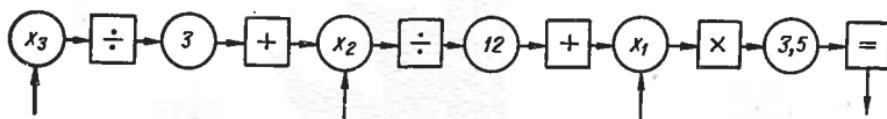
При многократных вычислениях по выбранному алгоритму, особенно при необходимости записи промежуточных результатов, необходимо использовать вычислительный бланк, выполняющий функции внешнего запоминающего устройства. Промежуточные результаты целесообразно записывать в бланк и в тех случаях, когда предложение на входном языке микрокалькулятора оказывается длинным и при ошибочном нажатии клавиши приходится полностью повторять вычисление. Однако регистрация в бланке большого количества промежуточных результатов также нежелательна, так как возрастает вероятность ошибки при регистрации чисел в бланке и их считывании. Оптимальная длина частных предложений, описывающих отдельные этапы вычислений, в основном определяется опытом и внимательностью пользователя.

Разбиение на вычислительные этапы может быть связано с лексикой микрокалькулятора используемого типа, так как при отсутствии операторов вычисления элементарных функций, входящих в расчетное выражение, их значения приходится брать из таблиц или вычислять при помощи различных разложений.

Форма вычислительного бланка, зависящая как от выбранного алгоритма вычислений, так и от особенностей входного языка микрокалькулятора, должна, по возможности, учитывать расчётные соотношения, последовательность выполняемых операций, исходные данные и фиксируемые промежуточные и окончательный результаты. Вероятность ошибочного нажатия клавишей, возрастающую при многократных вычислениях, можно уменьшить, дополнив бланк графикой вычислительного процесса с символами операций в кружках и символами вводимых операндов в прямоугольниках (табл. 9).

Грубых ошибок при вычислениях на микрокалькуляторах избегают, четко нажимая клавиши с одновременным контролем правильности их нажатия по индикатору и внимательно записывая числа в бланк и считывая их. Не следует слишком быстро нажимать

$$y = a \left((x_1 + \frac{x_2}{12} + \frac{x_3}{36}) = \left(\left(\frac{x_3}{3} + x_2 \right) : 12 + x_1 \right) a; a = 3,5 \right.$$



k	x_3	x_2	x_1	y
1	1,055	12,34	48,8	174,50173
2	6,25	14,41	32,4	118,21055
...

следующие клавиши, так как влияние переходных процессов в микровыключателях («дребезжание клавишей») и входных устройствах микрокалькулятора может привести к ошибочным результатам даже при правильном нажатии клавишей.

3. ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Моделирование результатов измерения физической величины x чаще всего сводится к статистической обработке (при многократных измерениях) множества (массива) исходных значений x_i и аппроксимации экспериментальных данных удобной для последующего использования буквенной моделью. Многие алгоритмы вычислений при таком моделировании удобно реализуются на микрокалькуляторах, причем при определении простейших статистических характеристик (оценок среднего и дисперсии) производительность инженерных микрокалькуляторов, снабженных дополнительным регистром памяти, практически не уступает производительности универсальных ЭВМ, так как в этих случаях вычисления связаны с вводом большого массива исходных данных при элементарном алгоритме вычислений. Затраты времени на ввод обрабатываемого массива данных сокращаются (а точность результата повышается) при переносе начала отсчета выборки из N обрабатываемых чисел x_i в точку x_0 , выбранную так, чтобы уменьшить число разрядов операндов.

Пример 12. Найти оценки среднего значения и дисперсии случайной величины x при ее выборке объемом $N = 5$: 1000,8; 1001,2; 998,5; 999,4; 1000,5.

В соответствии с формулами (13) непосредственно вычисляем $\bar{m}_1(x_i) = 5000,4 : 5 = 1000,08$. Для вычисления дисперсии (чтобы избежать необходимости вычитания оценки среднего из каждого значения x_i) целесообразно вначале вычислить второй начальный момент $\bar{m}_2 = 5000804,7 : 5 =$

= 1000160,9. Тогда, воспользовавшись связью начального и центрального моментов второго порядка, получим

$$\tilde{M}_2 = \tilde{m}_2 - \tilde{m}_1^2 = 1000160,9 - 1000160 = 0,9.$$

Погрешность значения \tilde{M}_2 можно оценить по соотношению

$$\frac{\delta \tilde{M}_2}{\tilde{M}_2} \leq \frac{\delta \tilde{m}_2}{\tilde{m}_2} \frac{\tilde{m}_2}{\tilde{M}_2} + \frac{\delta \tilde{m}_1^2}{\tilde{m}_1^2} \frac{\tilde{m}_1^2}{\tilde{M}_2},$$

где в первом приближении при вычислении на 8-разрядной микрокалькуляторе с представлением чисел в естественной форме можно принять

$$\delta \tilde{m}_2 / \tilde{m}_2 \leq N \cdot 10^{-7}; \delta \tilde{m}_1^2 / \tilde{m}_1^2 \leq N \cdot 10^{-7}.$$

Тогда $\delta \tilde{M}_2 / \tilde{M}_2 \leq 2 \cdot 5 \cdot 10^{-7} \cdot 10^6 : 0,9 \approx 1,11$, что заставляет сомневаться даже в первой цифре результата. Оценка дисперсии $\tilde{\sigma}_x^2 = 5 \cdot 0,9 : 4 = 1,125$, а среднеквадратичное отклонение $\sigma_x = 1,0606601$.

Если же принять в качестве начала отсчета $x_0 = 1000$, то потребуется ввести в микрокалькулятор разности $\delta x_j = x_j - x_0$, т. е. числа 0,8; 1,2; -1,5; -0,6; 0,5, что резко ускорит вычисления и повысит точность их результатов. Действительно, в этом случае среднее относительно $x_0 = 1000$ значение $\tilde{m}_1(\delta x_j) = 0,4 : 5 = 0,08$, второй начальный момент $\tilde{m}_2(\delta x_j) = 4,94 : 5 = 0,988$ и второй центральный $\tilde{M}_2(x_j) = 0,988 - 0,0064 = 0,9816$. В этом случае ошибка определяется как

$$\frac{\delta \tilde{M}_2}{\tilde{M}_2} \leq N \cdot 10^{-7} \frac{0,988}{0,9816} + N \cdot 10^{-7} \frac{0,0064}{0,9316} \approx 5 \cdot 10^{-7}.$$

Следовательно, более точное значение $\tilde{\sigma}_x^2 = 5 \cdot 0,9816 : 4 = 1,227$ отличается от ранее найденного уже во втором знаке.

При использовании микрокалькуляторов, имеющих хотя бы один регистр памяти, процесс одновременного вычисления среднего значения и дисперсии существенно упрощается. Особенно удобны в этом случае инженерные микрокалькуляторы (например, типа «Электроника Б3-18М»), имеющие оператор $\Pi + x^2$ суммирования квадратов чисел. Для таких микрокалькуляторов целесообразно составлять простые алгоритмы, позволяющие находить одновременно сумму вводимых операндов и сумму их квадратов ($\dots | Fx_i | \Pi + x^2 | + | Fx_{i+1} | \Pi + x^2 | + | \dots$). При вычислениях на арифметических микрокалькуляторах более двух моментов целесообразно составлять вычислительные бланки, в которые сначала заносить все необходимые степени переменной ($x_j - x_0$), а затем производить суммирование результатов по столбцам одинаковых степеней.

Значение функции t и коэффициентов q при определении доверительных интервалов по формулам (13в) и (13г) выбирают соответственно из табл. 1 и 2. Аналогично при проверке значимости коэффициентов корреляции, согласно формуле (13е), целесообразно использовать данные табл. 3.

L2

Пример 13. Найти среднее значение, дисперсию, их доверительные интервалы при $P_{\text{дов}} = 0,95$ и проверить «нормальность» закона распределения случайной величины x_j ; $j = 1, 2, \dots, 26$, выборка которых задана в табл. 10.

<i>I</i>	x_j	δx_j	$(\delta x_j)^2$	$(\delta x_j)^3$	$(\delta x_j)^4$
1	15,07	2,07	4,2849	8,869743	18,36036
2	13,47	0,47	0,2029	0,103823	0,0487968
3	15,65	2,65	7,0225	18,60962	49,31550
4	15,47	2,47	6,1009	15,06922	37,22098
5	16,06	3,06	9,3636	28,65261	87,67700
6	14,39	1,39	1,9321	2,685619	3,733010
7	14,13	1,13	1,2769	1,442897	1,630473
8	15,56	2,56	6,5536	16,77721	42,94967
9	13,67	0,67	0,4489	0,300763	0,2015112
10	14,31	1,31	1,7161	2,248091	2,944999
11	16,25	3,25	10,5625	34,32812	111,5664
12	17,24	4,24	17,9776	76,22502	323,1941
13	16,47	3,47	12,0409	41,78192	144,9832
14	16,66	3,66	13,3956	49,02789	179,4420
15	17,20	4,20	17,6400	74,08800	311,1696
16	14,93	1,93	3,72490	7,189057	13,87488
17	13,21	0,21	0,04410	0,009261	0,001945
18	16,18	3,18	10,1124	32,15743	102,2606
19	16,34	3,34	11,1556	37,25970	124,4474
20	14,84	1,84	3,38560	6,229504	11,46228
21	14,93	1,93	3,72490	7,189057	13,87488
22	13,84	0,84	0,70560	0,592704	0,497871
23	15,30	2,30	5,29000	12,16700	27,98410
24	13,99	0,99	0,98010	0,970299	0,960596
25	16,49	3,49	12,1801	42,50854	148,3548
26	13,35	0,35	0,12250	0,042875	0,015006
$\sum (\delta x_j)^k$		—	57.	161,9628	516,52602
					1758,1722

Выбрав начало отсчета $x_0 = 13$, заносим в вычислительный бланк (табл. 10) значения $\delta x_j = x_j - x_0$ и вычисляем степени δx_j до четвертой включительно. После суммирования чисел в столбцах, согласно формуле (13а), последовательно находим оценки

$$\tilde{m}_1(\delta x_j) = 57 : 26 = 2,192307; \quad \tilde{m}_2(\delta x_j) = 161,9628 : 26 = 6,229338;$$

$$\tilde{m}_3(\delta x_j) = 516,52602 : 26 = 19,86638; \quad \tilde{m}_4(\delta x_j) = 1758,1722 : 26 = 67,62200.$$

Используя связь центральных и начальных моментов, вычисляем

$$\tilde{M}_2(\delta x_j) = \tilde{m}_2 - \tilde{m}_1^2 = 1,423128; \quad \tilde{M}_3(\delta x_j) = \tilde{m}_3 - 3\tilde{m}_1\tilde{m}_2 + 2\tilde{m}_1^3 = -0,030109;$$

$$\tilde{M}_4(\delta x_j) = \tilde{m}_4 - 4\tilde{m}_1\tilde{m}_3 + 6\tilde{m}_1^2\tilde{m}_2 - 3\tilde{m}_1^4 = 3,747207.$$

Тогда $m_1(x_j) = m_1(\delta x_j) + x_0 = 15,192307; \quad \tilde{\sigma}_x^2 = \tilde{M}_2(\delta x_j) N/(N-1) = 1,480053; \quad \tilde{\gamma}_3 = \tilde{M}_3/\sqrt{\tilde{M}_2^3} = -0,0177343; \quad \tilde{\gamma}_4 = -3 + \tilde{M}_4/\tilde{M}_2^2 = -1,149795.$

Для определения доверительных интервалов при $P_{\text{дов}} = 0,95$ и $N = 26$ по табл. 1 и 2 находим $t = 2,060$ и $q = 0,302$. Следовательно, $|\tilde{m}_1 - m_1| < 2,06 \times 1,216574/\sqrt{26} = 0,4914949$; $|\tilde{\sigma}_x - \sigma_x| < 0,302\sqrt{1,480053}$ и, после округления,

$$m_1(x_j) \in [14,7; 15,7]; \quad \sigma_x \in [0,85; 1,58].$$

Для проверки близости закона распределения к нормальному, согласно формуле (13д), вычисляем вспомогательные коэффициенты

$$\sqrt{6(N-1)/(N+1)(N+3)} = 0,437688;$$
$$\sqrt{24N(N-2)(N-3)/(N-1)^2(N+3)(N+5)} = 0,7829643.$$

Так как $\tilde{\gamma}_3$ и γ_4 не превышают вычисленные вспомогательные коэффициенты соответственно более чем в два раза, то закон распределения можно принять нормальным.

При вычислениях эмпирического коэффициента корреляции между двумя случайными величинами, согласно формуле (13е), можно использовать любое линейное преобразование исходных данных, так как коэффициент корреляции инвариантен к подобным преобразованиям.

При моделировании экспериментально измеренной зависимости $y(x)$ табличной моделью $y_i = y(x_i)$; $i = 0, 1, 2, \dots, n$, полученной с некоторой погрешностью, часто используют метод наименьших квадратов. Простейшей является линейная зависимость между переменными x и y , определяемая коэффициентами регрессии и обеспечивающая минимум средквадратичной ошибки.

Коэффициенты линейной регрессии связаны с коэффициентом корреляции соотношениями

$$\tilde{\beta}_{y/x} = \tilde{r}\tilde{\sigma}_y/\tilde{\sigma}_x; \quad \tilde{\beta}_{x/y} = \tilde{r}\tilde{\sigma}_x/\tilde{\sigma}_y$$

и для их определения проще всего воспользоваться соотношением, подобным соотношению

$$\tilde{\beta}_{x/y} = \frac{\sum_{i=1}^n x_i y_i - N \tilde{m}_{1x} \tilde{m}_{1y}}{\sum_{i=1}^n y_i^2 - N (\tilde{m}_{1y})^2} = \frac{\sum_{i=1}^n u_i v_i - N \tilde{m}_{1u} \tilde{m}_{1v}}{\sum_{i=1}^n v_i^2 - N (\tilde{m}_{1v})^2} \frac{h_1}{h_2},$$

где h_1 и h_2 — нормирующие коэффициенты, связывающие $u = (x - x_0)/h_1$ и $v = (y - y_0)/h_2$.

Пример 14. Определить коэффициенты корреляции и линейной регрессии для массива случайных чисел x и y , заданных парами значений $(x_j; y_j) = (12,5; 0,03), (10,4; 0,15), (14,8; 0,25), (9,3; 0,01), (12,2; 0,04), (14,8; 0,32), (9,3; 0,03), (11,6; 0,09), (12,3; 0,05), (13,4; 0,07)$.

Приняв вместо x_j величину $u_j = 10(x_j - 10)$, а вместо y_j величину $v_j = 100y_j$, получим пары значений $(u_j, v_j) = (25; 3), (4; 15), (48; 25), (-7; 1), (22; 4), (48; 32), (-7; 3), (16; 9), (23; 5), (34; 7)$. Тогда $\sum u_j v_j = 3428$; $m(u_j) = 20,6$; $m(v_j) = 10,4$; $\sum u_j^2 = 7772$; $\sum v_j^2 = 2064$ и, согласно формуле (13е), коэффициент корреляции $r = 0,6905144$.

Вычислив $H = 0,6905144 \sqrt{10-1} = 2,0715432$, по табл. 3 для $P_{\text{дов}} = 0,95$ и $N = 10$ находим критическое значение 1,9. Так как вычисленное значение H больше критического, то заданные величины действительно коррелированы.

В рассматриваемом случае $h_1 = 0,1$ и $h_2 = 0,01$, а коэффициенты регрессии

$$\tilde{\beta}_{x/y} = \frac{(3428 - 10 \cdot 10,4 \cdot 20,6) \cdot 0,1}{(2064 - 10 \cdot 10,4 \cdot 10,4) \cdot 0,01} = 13,086319;$$

$$\tilde{\beta}_{y/x} = \frac{(3428 - 10 \cdot 10,4 \cdot 20,6) \cdot 0,01}{(7772 - 10 \cdot 20,6 \cdot 20,6) \cdot 0,1} = 0,03643577.$$

Следовательно, уравнения прямых регрессии

$$x - \tilde{m}_{1x} = \tilde{\beta}_{x/y} (y - \tilde{m}_{1y}); \quad x = 13,086319y + 10,839023;$$

$$y - \tilde{m}_{1y} = \tilde{\beta}_{y/x} (x - \tilde{m}_{1x}); \quad y = 0,03643577x - 0,3350907.$$

Однако решение подобных задач при большом N оказывается затруднительным еще на этапе выбора аппроксимирующей функции. Кроме того, на практике часто встречаются задачи, где прежде всего необходимо «очистить» результаты измерений от «шума», вызванного погрешностями измерений. В этих случаях целесообразно использовать методы сглаживания найденных экспериментально зависимостей, легкореализуемые при помощи микрокалькуляторов.

При линейном сглаживании по трем очередным значениям y_i табличной модели с постоянным шагом используют формулу [21]

$$\tilde{y}_i = (y_{i-1} + y_i + y_{i+1})/3, \quad (111a)$$

причем для краев таблицы используют соотношения

$$\tilde{y}_0 = (5y_0 + 2y_1 - y_2)/6; \quad \tilde{y}_n = (5y_n + 2y_{n-1} + y_{n-2})/6.$$

Линейное сглаживание по пяти точкам реализуется по формуле

$$\tilde{y}_i = (y_{i-2} + y_{i-1} + y_i + y_{i+1} + y_{i+2})/5, \quad (111b)$$

но крайние точки вычисляют по формулам

$$\tilde{y}_0 = (3y_0 + 2y_1 + y_2 - y_4)/5; \quad \tilde{y}_1 = (4y_0 + 3y_1 + 2y_2 + y_3)/10;$$

$$\tilde{y}_{n-1} = (y_{n-3} + 2y_{n-2} + 3y_{n-1} + 4y_n)/10;$$

$$\tilde{y}_n = (3y_n + 2y_{n-1} + y_{n-2} - y_{n-4})/5.$$

При нелинейном сглаживании (многочленом третьей степени, по семи точкам

$$\begin{aligned} \tilde{y}_i = & (-2y_{i-3} + 3y_{i-2} + 6y_{i-1} + 7y_i + 6y_{i+1} + \\ & + 3y_{i+2} - 2y_{i+3})/21 \end{aligned} \quad (111b)$$

и на краях таблицы

$$\tilde{y}_0 = (39y_0 + 8y_1 - 4y_2 - 4y_3 + y_4 + 4y_5 - 2y_6)/42;$$

$$\tilde{y}_1 = (8y_0 + 19y_1 + 16y_2 + 6y_3 - 4y_4 - 7y_5 + 4y_6)/42;$$

$$\tilde{y}_2 = (-4y_0 + 16y_1 + 19y_2 + 12y_3 + 2y_4 - 4y_5 + y_6)/42;$$

$$\begin{aligned} \tilde{y}_{n-2} = & (y_{n-6} - 4y_{n-5} + 2y_{n-4} + 12y_{n-3} + 19y_{n-2} + \\ & + 16y_{n-1} - 4y_n)/42; \end{aligned}$$

$$\tilde{y}_{n-1} = (4y_{n-6} - 7y_{n-5} + 4y_{n-4} + 6y_{n-3} + 16y_{n-2} + \\ + 19y_{n-1} + 8y_n)/42;$$

$$\tilde{y}_n = (-2y_{n-6} + 4y_{n-5} + y_{n-4} - 4y_{n-3} - 4y_{n-2} + \\ + 8y_{n-1} + 39y_n)/42.$$

Сглаживание улучшается при его повторении для полученных результатов.

После статистической обработки точность табличной модели определяется доверительными интервалами значений функции $y_i = y(x_i)$ в узлах модели. Значения функции в интервалах между узлами x_i остаются неизвестными и лишь на основании физических предположений о поведении функции в этих интервалах можно достаточно обоснованно выбрать способ интерполяции.

Для интерполирования табличной модели с равномерным шагом $h = x_{i+1} - x_i$ обычно используют формулы Ньютона (17). Если заданное значение аргумента x находится ближе к началу таблицы, то используют многочлен (17а), который для вычислений на микрокалькуляторе удобно записать как

$$P(x) = \left(((\dots + \Delta^3 y_i) \frac{q-2}{3} + \Delta^2 y_i) \frac{q-1}{2} + \Delta y_i \right) q + y_i, \quad (112a)$$

где i — номер ближайшего к $x > x_i$ узла интерполяции; $q = (x - x_i)/h$.

Если заданное значение аргумента ближе к концу таблицы, то используют многочлен (17б), записываемый для вычислений на микрокалькуляторе в виде формулы

$$P(x) = (((\dots + \Delta^3 y_{i-3}) \frac{q+2}{3} + \Delta^2 y_{i-2}) \frac{q+1}{2} + \Delta y_{i-1}) q + y_i, \quad (112b)$$

где i — номер ближайшего к $x < x_i$ узла интерполяции.

Остаточная погрешность этих формул при их усечении по отношению к многочлену m -й степени определяется выражением*

$$R_m(x) \approx \frac{q(q \mp 1)\dots(q \mp m)}{(m+1)!} \Delta^{(m+1)} y_i. \quad (112b)$$

При использовании многочлена Ньютона для обратной интерполяции с помощью микрокалькулятора формулу (20) удобно записать как

$$q_{k+1} = \left(\left((\dots + \Delta^3 y_i) \frac{q_k-2}{3} + \Delta^2 y_i \right) \frac{-q_k+1}{2} - y_i + y \right) / \Delta y_i, \quad (112c)$$

используя в качестве нулевого приближения $q_0 = (y - y_i) / \Delta y_i$.

* Знаки «—» и «+» относятся к формулам (112a) и (112b) соответственно.

Пример 15. По результатам измерения физической зависимости, заданным табличной моделью (табл. 5), определить значения функции $y(x)$ в точках $x = 0,15; 0,85; -0,1$, а также значение аргумента при $y = 4$.

Для интерполяции табличной модели при $x = 0,15$ определяем $x_i = x_1 = 0,1$ и $q = 0,5$. По формуле (112в) и приведенным в табл. 5 конечным разностям вычисляем $R_1(0,15) = 0,5(0,5 - 1)(-0,2)/2 = 0,025$. Так как исходные данные заданы с точностью до двух значащих цифр, то требуемая точность вычисления $y(x)$ не превышает 0,05 и можно использовать линейную интерполяцию. По формуле (112а) при $m = 1$ вычисляем $y(0,15) = 6,3 + 0,5(-1,1) = 5,75 = 5,8$ с точностью до двух значащих цифр. При квадратичной интерполяции $y(0,15) = 6,3 + 0,5(-1,1 - 0,5)(-0,2)/2 \approx 5,8$ с остаточной погрешностью $R_2(0,15) = 0,5(0,5 - 1)(0,5 - 2)(-0,3)/3! = -0,01875$.

Для $x = 0,85$ принимаем $x_i = x_3 = 0,9$; $q = -0,5$ и вычисляем $R_1(0,85) = -0,5(-0,5 + 1)(-0,7)/2 = 0,0875$, что превышает допустимую ошибку (равную 0,05) при линейной интерполяции. Если использовать квадратичную интерполяцию, то ошибка близка к нулю, так как $\Delta^3y(0,9) = 0$. В этом случае $y(0,85) = 5,5 - 0,5(-0,3 + 0,5(-0,7)/2) = 5,7375 \approx 5,7$.

Точность экстраполяции функции уменьшается при увеличении интервала экстраполяции. Приняв $x_i = x_0 = 0$; $q = -1$, для обеспечения максимальной точности последовательно вычисляем $R_1(-0,1) = -1(-2)(-0,4)/2 = -0,4$; $R_2(-0,1) = -1(-2)(-3)(0,2)/3! = -0,2$; $R_3(-0,1) = -1(-2)(-3)(-4)(-0,5)/4! = -0,5$. Сравнивая полученные значения остаточных погрешностей, выбираем квадратичную экстраполяцию, обеспечивающую минимальную ошибку, и вычисляем $y(-0,1) = 7 - (0,7 + (-1 - 1)(-0,4)/2) = 7,3$.

Для обратного интерполирования находим интервалы аргумента $(0,2; 0,3)$ и $(0,5; 0,6)$, соответствующие значениям функции $y(x) = 4$. Согласно выражению (112г), при $m = 2$ составляем для первого интервала расчетную формулу

$$q_{k+1} = (y - y_2 - 0,5(q_k - 1)\Delta^2y_2)/\Delta y_2 = 0,923 - (q_k - 1)0,192.$$

Приняв в качестве нулевого приближения $q_0 = 0,923$ для $k = 1, 2, \dots$, последовательно находим $q_k = 0,938; 0,935; 0,935$. Следовательно, $x(4) = qh + x_2 = 0,935 \cdot 0,1 + 0,2 = 0,29$.

Аналогично для второго интервала составляем формулу

$$q_{k+1} = (y - y_5 - 0,5(q_k - 1)\Delta^2y_5)/\Delta y_5 = 0,786 + 0,107(q_k - 1)$$

и, приняв $q_0 = 0,786$, находим $q_k = 0,763; 0,761; 0,760; 0,760$. Следовательно, в этом интервале $x(4) = 0,76 \cdot 0,1 + 0,5 = 0,58$.

Выбор буквенного выражения, приближающего табличную модель, определяется не только требуемой точностью аппроксимации, но и удобством последующего использования буквенной модели для вычислений на микрокалькуляторе. Поэтому расчетная формула должна содержать только символы операторов, содержащихся во входном языке микрокалькулятора используемого типа, и обеспечивать цепные вычисления с минимальным числом запоминаемых промежуточных результатов.

При использовании микрокалькуляторов с недостаточной емкостью запоминающих устройств существенное значение имеет число знаков в коэффициентах буквенной модели, так как для ввода каждого десятичного знака необходимо затратить столько же времени, сколько и для ввода арифметического оператора. Поэтому для таких микрокалькуляторов эффективность расчетных формул при

сохранении требуемой точности аппроксимации следует оценивать по числу всех операторов (нажатий клавишей), а не по числу операций.

Для вычислений на микрокалькуляторе любого типа достаточно удобны аппроксимации табличных моделей степенными многочленами, записанными в виде формулы (33), обеспечивающей цепные операции без запоминания промежуточных результатов. Однако при использовании инженерных микрокалькуляторов, автоматически вычисляющих элементарные функции, затраты времени часто удается уменьшить, используя буквенные модели, содержащие операторы вычисления таких функций.

Удобные для вычислений буквенные модели часто удается составить посредством округления коэффициентов более сложных формул или таким подбором коэффициентов с небольшим числом десятичных знаков, при котором обеспечивается требуемая точность аппроксимации. Для выбора вида аппроксимирующей функции часто полезно предварительно построить графическую модель.

Следует учитывать, что затраты времени на составление простой и достаточно точной расчетной формулы с избытком компенсируются при многократном использовании этой формулы для вычислений.

Пример 16. Составить формулу, аппроксимирующую с точностью не менее 5 % табличную модель $y_i = 2,0; 2,5; 3,6; 4,9; 6,8; 9,0$ в интервале аргумента $0 \leq x \leq 10$ с постоянным шагом $h = 2$.

Попытаемся выполнить условия задачи выбором одной из аппроксимирующих функций, приведенных в табл. 7. Выбрав значения $x_a = x_0 = 0$; $y(x_a) = 2$; $x_b = x_5 = 10$; $y(x_b) = 9$, вычисляем $x_a = 5$; $x_r = x_{rp} = 0$; $y_a = 5,5$; $y_r = 4,243$; $y_{rp} = 3,273$; $y(x_r) = y(x_{rp}) = 2$. Линейным интерполированием находим

$$y(x_a) = 3,6 + (4,9 - 3,6)/2 = 4,25$$

и вычисляем разности модулей $|y(x_a) - y_a| = 1,25$; $|y(x_a) - y_r| = 0,007$; $|y(x_r) - y_r| = 2,243$; $|y(x_r) - y_a| = |y(x_{rp}) - y_a| = 3,5$; $|y(x_a) - y_{rp}| = 0,977$; $|y(x_{rp}) - y_{rp}| = 1,273$.

В соответствии с данными табл. 7 выбираем аппроксимирующую функцию $f(x) = ab^x$. Для точек $x_0 = 0$ и $x_4 = 8$ составляем систему уравнений $ab^0 = 2$; $ab^8 = 6,8$, после решения которой получаем расчетную формулу $f(x) = 2 \cdot 1,165292^x$ с относительной погрешностью более 8 %. Подбирая коэффициенты, получаем формулу $f(x) = 1,9 \cdot 1,17^x$ с максимальной погрешностью не более 5 % во всех узлах заданной табличной модели.

Если в используемом микрокалькуляторе входной язык не содержит оператора вычисления степенной функции, то полученная формула малопригодна, так как должна быть дополнитель но преобразована к удобному для вычислений виду. Поэтому попробуем аппроксимировать заданную табличную модель многочленом $a_0 + a_1x + a_2x^2$. Выбрав $x_0 = 0$; $x_2 = 4$; $x_6 = 10$, составляем по методу неопределенных коэффициентов систему уравнений: $y(0) = a_0$; $y(4) = a_0 + 4a_1 + 16a_2$; $y(10) = a_0 + 10a_1 + 100a_2$. Решив эту систему, получаем удобную для многократных вычислений формулу

$$y(x) \approx (x + 4)0,05x + 2 = (x + 4)x/20 + 2$$

с относительной погрешностью не более 5 % во всех узлах заданной табличной модели.

При вычислениях по полученной формуле необходимо дважды вводить значение аргумента, что при большом числе знаков в его представлении

потребует дополнительных затрат времени. Поэтому попытаемся составить формулу, требующую однократного введения аргумента. Выбрав функцию с тремя коэффициентами

$$f(x) = a + b/(c + x),$$

составим для узлов модели со значениями аргументов $x_0 = 0$ (в этом узле относительная погрешность максимальна), $x_2 = 4$ и $x_4 = 8$ систему уравнений в соответствии с методом неопределенных коэффициентов. После решения этой системы уравнений относительно искомых коэффициентов и округления и подбора их значений, получим окончательно расчетную формулу

$$y(x) \approx 96/(17,5 - x) - 3,58$$

с относительной погрешностью менее 5% для всех узлов исходной модели. Эта формула при запоминании результата вычитания $17,5 - x$ требует только трех арифметических операций, а при использовании микрокалькулятора с входным языком, содержащим оператор $1/x$, только трех цепных арифметических операций, реализуемых вводом одного предложения.

Аппроксимация табличных моделей с $n + 1$ узлами многочленами n -й степени упрощается, если предварительно составить аналитические выражения для вычисления коэффициентов многочлена без решения систем уравнений вида (29). С этой целью можно использовать интерполирующий многочлен (17а), записав его как

$$P(z) = \sum_{k=0}^n b_k z^{[k]} = \sum_{k=0}^n \frac{\Delta^k y_0}{k! h^k} z^{[k]}, \quad (113a)$$

где $z^{[k]} = z(z - h) \dots (z - (k - 1)h)$ — обобщенная степень [8] числа $z = x - x_0$.

Суммируя члены этого разложения с одинаковыми степенями x , получим формулы, связывающие коэффициенты a_i аппроксимирующего многочлена с конечными разностями модели $y_i = y(x_i)$ с постоянным шагом. Подобные формулы записывают в удобном для вычислений на микрокалькуляторе виде как

$$a_0 = y_0;$$

$$a_1 = \left(\left(\left(\left(\dots + \Delta^5 y_0 \right) \frac{4}{5} - \Delta^4 y_0 \right) \frac{3}{4} + \Delta^3 y_0 \right) \frac{2}{3} - \Delta^2 y_0 \right) \frac{1}{2} + \Delta y_0 \right) \frac{1}{h};$$

$$a_2 = \left(\left(\left(\dots - \Delta^5 y_0 \right) \frac{10}{11} + \Delta^4 y_0 \right) \frac{11}{12} - \Delta^3 y_0 + \Delta^2 y_0 \right) \frac{1}{2h^2};$$

$$a_3 = \left(\left(\left(\dots + \Delta^5 y_0 \right) \frac{7}{6} - \Delta^4 y_0 \right) \frac{3}{2} + \Delta^3 y_0 \right) \frac{1}{6h^3};$$

$$a_4 = \left(\left(\dots - 2\Delta^5 y_0 + \Delta^4 y_0 \right) \frac{1}{24h^4}; \quad (113b)$$

$$a_5 = \left(\dots + \Delta^5 y_0 \right) \frac{1}{120h^5}; \dots$$

Каждый коэффициент a_i при расчете по этим формулам вычисляют на микрокалькуляторе вводом одного предложения.

Если в табличной модели $x_0 \neq 0$, то коэффициенты аппроксимирующего многочлена $P(z) = P(x - x_0)$ вычисляют подстановкой

полученных по формулам (113б) коэффициентов в обобщенную схему Горнера с делителем x_0 (см. пример 7). Практически важное значение в ряде инженерных приложений имеет случай, когда начало отсчета аргумента $z = x - 0,5 (x_n - x_0)$ расположено в центре интервала аргумента x табличной модели. В этом случае коэффициенты аппроксимирующего многочлена $P(z)$ можно вычислить без предварительного вычисления конечных разностей, воспользовавшись

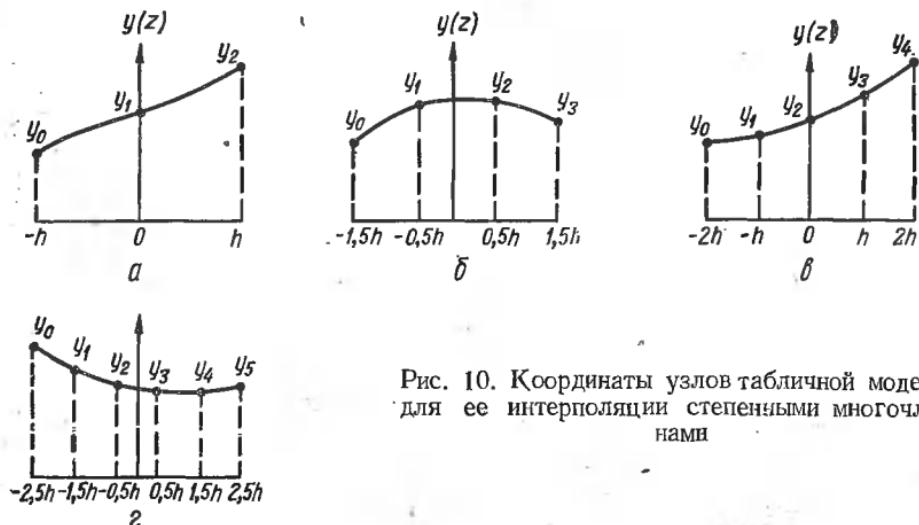


Рис. 10. Координаты узлов табличной модели для ее интерполяции степенными многочленами

формулами, выражающими коэффициенты многочлена через значения таблично заданной функции.

Подобные формулы для многочлена второй степени (рис. 10, а):

$$\begin{aligned} a_2 &= (-2y_1 + y_2 + y_0)/2h^2; \\ a_1 &= (y_2 - y_0)/2h; \quad a_0 = y_1, \end{aligned} \tag{114a}$$

для многочлена третьей степени (рис. 10, б):

$$\begin{aligned} a_3 &= ((y_3 - y_0) : 3 + y_1 - y_2)/2h^3; \\ a_2 &= (y_3 + y_1 - y_2 - y_0)/4h^2; \\ a_1 &= (-a_3h^3 : 4 + y_2 - y_1)/h; \\ a_0 &= (-a_2h^2 : 2 + y_2 + y_1)/2, \end{aligned} \tag{114б}$$

для многочлена четвертой степени (рис. 10, в):

$$\begin{aligned} a_4 &= ((1.5y_2 - y_1 - y_3)4 + y_4 + y_0)/24h^4; \\ a_3 &= ((y_1 - y_3)2 + y_4 - y_0)/12h^3; \\ a_2 &= (-2y_2 + y_3 + y_1) : h^2 - a_4h^2; \\ a_1 &= (y_1 - y_3) : 2h + a_3h^2; \\ a_0 &= y_2, \end{aligned} \tag{114в}$$

для многочлена пятой степени (рис. 10, г):

$$\begin{aligned}
 a_5 &= (((y_3 - y_2)2 + y_1 - y_4)5 + y_5 - y_0)/120h^5; \\
 a_4 &= (((y_4 + y_1)(-1,5) + y_3 + y_2)2 + y_5 + y_0)/48h^4; \\
 a_3 &= ((-5a_5h^5 + y_2 - y_3)3 + y_4 - y_1)/6h^3; \\
 a_2 &= ((-10a_4h^4 + y_4 + y_1 - y_3 - y_2)/4h^2; \\
 a_1 &= ((-a_5h^2 : 4 - a_3h^3 : 4 + y_3 - y_2)/h; \\
 a_0 &= ((-a_4h^2 : 4 - a_2)h^2 : 2 + y_3 + y_2)/2. \quad (114\text{г})
 \end{aligned}$$

При необходимости перехода от многочлена $P(z)$, коэффициенты которого определены по формулам (114), к многочлену $P(x)$, соответствующему совпадению начала отсчета с узлом x_0 , достаточно использовать обобщенную схему Горнера для $z = x - 0,5$ ($x_n - x_0$).

Пример 17. Аппроксимировать табличную модель, заданную в табл. 11, степенным многочленом и оценить погрешность аппроксимации.

Таблица 11

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$
0	0	8,0	0,3	3,8	-2,2	-3,0	5,5
1	1	8,3	4,1	1,6	-5,2	2,5	
2	2	12,4	5,7	-3,6	-2,7		
3	3	18,1	2,1	-6,3			
4	4	20,2	-4,2				
5	5	16,0					

Вычисляя конечные разности табличной модели (табл. 11), подставляем их значения в формулы (113б) и при $h = 1$ и $z = x - x_0 = x$ находим интерполирующий многочлен

$$P(x) = 8 - 0,483333x - 0,666666x^2 + 1,9875x^3 - 0,583333x^4 + 0,045833x^5.$$

Подстановка в многочлен значения x_i для узлов заданной модели подтверждает практически полное отсутствие погрешности.

При аппроксимации табличной модели многочленом пятой степени с началом отсчета в середине интервала аргумента (см. рис. 10, а) по формулам (114г) находим

$$\begin{aligned}
 P(z) = 15,36914 + 5,942448z - 0,473958z^2 - 0,98125z^3 - 0,010417z^4 + \\
 + 0,045833z^5,
 \end{aligned}$$

где $z = x - 0,5(x_n - x_0)$. Используя для переноса начала отсчета обобщенную схему Горнера с вычислениями по формуле (33), находим коэффициенты многочлена $P(x)$ с коэффициентами найденного интерполирующего многочлена с точностью до округляемых знаков.

Аппроксимирующие многочлены с меньшим числом коэффициентов составляют по критерию минимального среднеквадратичного отклонения в соответствии с формулой (28а). Для линейной аппроксимации заданной табличной модели методом наименьших квадратов составляем систему уравнений

$$\begin{bmatrix} 6 & 15 \\ 15 & 55 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 83 \\ 248,2 \end{bmatrix},$$

коэффициенты которой вычисляем в соответствии с табл. 12. Решив систему относительно искомых коэффициентов, составляем аппроксимирующий многочлен первой степени

$$P(x) = 8,019047 + 2,325714x$$

Таблица 12

i	x_i	y_i	x_i^2	x_i^3	x_i^4	x_i^5	x_i^6	$y_i x_i$	$y_i x_i^2$	$y_i x_i^3$
0	0	8	0	0	0	0	0	0	0	0
1	1	8,3	1	1	1	1	1	8,3	8,3	8,3
2	2	12,4	4	8	16	32	64	24,8	49,6	99,2
3	3	18,1	9	27	81	243	729	54,3	162,9	488,7
4	4	20,2	16	64	256	1024	4096	80,8	323,2	1292,8
5	5	16	25	125	625	3125	15625	80	400	2000
Σ	15	83	55	225	979	4425	20515	248,2	944	3889

с максимальной абсолютной погрешностью $\epsilon_a = 3,65$ (при $x = 5$) и максимальной относительной погрешностью менее 25% (при $x = 1$).

Используя данные табл. 12, составляем согласно формуле (28в) систему уравнений

$$\begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 83 \\ 248,2 \\ 944 \end{bmatrix},$$

решение которой дает аппроксимирующий многочлен второй степени

$$P(x) = 6,203572 + 5,048928x - 0,556428x^2,$$

с максимальными абсолютной погрешностью около 2,515 (при $x = 4$) и относительной погрешностью менее 29% (при $x = 2$).

По вычисленным в табл. 12 данным составляем систему уравнений

$$\begin{bmatrix} 6 & 15 & 55 & 225 \\ 15 & 55 & 225 & 979 \\ 55 & 225 & 979 & 4425 \\ 225 & 979 & 4425 & 20515 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 83 \\ 248,2 \\ 944 \\ 3889 \end{bmatrix}.$$

Решая эту систему уравнений, находим многочлен

$$P(x) = 8,039651 - 3,335870x + 4,045579x^2 - 0,6120298x^3,$$

максимальная абсолютная погрешность которого в узлах модели составляет около 0,25 при $x = 2$, а максимальная относительная погрешность — менее 2% (при $x = 2$).

Составленный методом наименьших квадратов аппроксимирующий многочлен пятой степени, очевидно, совпадет с ранее составленным интерполирующими многочленом с точностью до погрешностей округления. Если по этому многочлену определить значения y_i для аппроксимации многочленом четвертой степени (см. рис. 10, б), то последний будет иметь большую погрешность по сравнению с многочленом третьей степени, составленным методом наименьших квадратов.

Графики значений многочленов первой, второй и пятой степени в заданном интервале аргумента показаны для сравнения на рис. 11.

Повышение степени аппроксимирующего многочлена не всегда приводит (о чем свидетельствуют и данные рассмотренного примера для многочлена второй степени) к уменьшению относительной

погрешности, зависящей от значения функции. Поэтому при необходимости минимизировать относительную погрешность аппроксимации методом наименьших квадратов вместо формулы (28а) следует использовать систему уравнений

$$\sum_{k=0}^m a_k \sum_{i=0}^n \frac{x_i^{k+s}}{f(x_i)^2} = \sum_{i=0}^n \frac{x_i^s}{f(x_i)}; \quad s = 0, 1, \dots, n, \quad (115)$$

где m — степень аппроксимирующего многочлена; $n + 1$ — число точек модели (включая начальную), выбранных для приближения.

При необходимости минимизации максимальных значений абсолютной погрешности аппроксимации можно рекомендовать следующий алгоритм:

1. После определения аппроксимирующего многочлена методом наименьших квадратов вычислить погрешности аппроксимации $\delta_i = y_i - P(x_i)$ для всех узлов заданной табличной модели.

2. Из полученного множества δ_i выбрать $m + 2$ знакочередующихся экстремальных значений $\delta_i = \alpha_j$; $i = j = 0, 1, \dots, m + 2$, что всегда возможно при отсутствии ошибок в составлении аппроксимирующего многочлена методом наименьших квадратов.

Рис. 11. Графики значений аппроксимирующих многочленов первой, второй и пятой степеней

многочлена методом наименьших квадратов.

3. Составить систему из $m + 2$ уравнений

$$\pm [P(x_j) - y_j] = \alpha; \quad j = 0, 1, \dots, m + 2,$$

где искомыми являются значения коэффициентов аппроксимирующего многочлена и минимальный модуль наибольших отклонений α .

4. После решения составленной системы уравнений проверить погрешности δ_i во всех узлах модели. Если окажется, что в одном или нескольких узлах погрешность превышает α , то сформировать новую систему уравнений и повторить процедуру до получения одинаковых погрешностей в узлах модели.

Пример 18. Минимизировать среднеквадратичную относительную и максимальную абсолютную погрешность аппроксимирующих многочленов первой и второй степени, составленных в предыдущем примере.

Составив таблицу значений $x_i^s/f(x_i)^2$ и $x_i^s/f(x_i)$ и вычислив их суммы согласно формуле (115), формируем систему уравнений

$$\begin{bmatrix} 4,605393 & 6,601461 \\ 6,601461 & 20,487202 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 49,33806 \\ 95,80378 \end{bmatrix}.$$

Решив эту систему уравнений, получим аппроксимирующую многочлен первой степени $P(x) = 7,451911 + 2,275112x$ с максимальной относительной погрешностью 21,12% ($x = 3$) при абсолютной погрешности 3,82 в том же узле.

Аналогично для многочлена второй степени, решив уравнения

$$\begin{bmatrix} 4,605393 & 6,601461 & 20,48702 \\ 6,601461 & 20,48702 & 79,40887 \\ 20,48702 & 79,40887 & 343,4615 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 49,33806 \\ 95,80378 \\ 329,4879 \end{bmatrix},$$

получим многочлен

$$P(x) = 7,137609 + 2,968296x - 0,1527086x^2$$

с максимальной относительной погрешностью 19,92% при $x = 1$ и абсолютной максимальной погрешностью 3,63 при $x = 4$.

Вычислив для составленного в примере 17 многочлена первой степени абсолютные погрешности в узлах модели $\delta_i = P(x_i) - y_i$, выберем значения аргумента $x_1 = 1$, $x_2 = 3$ и $x_3 = 5$ с экстремальными отклонениями $\delta_1 = 2,04$; $\delta_2 = -3,1$; $\delta_3 = 3,65$. Составим систему уравнений

$$\begin{aligned} a_1 x_1 + a_0 - 8,3 &= \alpha; \\ -(a_1 x_2 + a_0 - 18,1) &= \alpha; \\ a_1 x_3 + a_0 - 16 &= \alpha, \end{aligned}$$

решение которой дает $a_1 = 1,925$; $a_0 = 9,35$; $\alpha = 2,975$ или $P(x) = 9,35 + 1,925x$.

Определив для этого многочлена значения погрешностей в узлах модели $\delta_1 = 1,35$; $2,975$; $0,8$; $-2,975$; $-3,15$; $2,975$, составим систему уравнений

$$\begin{aligned} a_1 + a_0 - 8,3 &= \alpha; \\ -(4a_1 + a_0 - 20,2) &= \alpha; \\ 5a_1 + a_0 - 16 &= \alpha, \end{aligned}$$

решение которой дает аппроксимирующее выражение $P(x) = 9,4375 + 1,925x$ с максимальной абсолютной погрешностью $\alpha = 3,0625$ для значений $x = 1; 4; 5$.

Аналогично для многочлена второй степени, составленного в предыдущем примере, максимальные отклонения будут соответствовать значениям $x = 0; 1; 4; 5$. Система корректирующих уравнений в этом случае будет иметь вид

$$\begin{aligned} \alpha &= a_0 - 8; \\ \alpha &= -(a_0 + a_1 + a_2 - 8,3); \\ \alpha &= a_0 + 4a_1 + 16a_2 - 20,2; \\ \alpha &= -(a_0 + 5a_1 + 25a_2 - 16). \end{aligned}$$

Решив эту систему уравнений, получим аппроксимирующий многочлен

$$P(x) = 5,78125 + 5,3x - 0,5625x^2$$

с максимальной абсолютной погрешностью $\alpha = 2,21875$ для значений $x = 0; 1; 4; 5$.

Для аппроксимации можно использовать различные функции, но при использовании микрокалькуляторов любого типа особенно удобны степенные ряды. Методы аппроксимации используют не только для составления буквенной модели по заданной табличной, но и при необходимости преобразования заданной буквенной модели к другому (обычно более удобному для вычислений) виду. В этих случаях, кроме возможного аналитического преобразования модели, можно использовать построение табличной модели по заданной буквенной с последующей ее аппроксимацией любым методом.

4. ВЫЧИСЛЕНИЕ ФУНКЦИЙ ПО БУКВЕННЫМ МОДЕЛЯМ

Способ вычисления функции $y(x)$ для заданного значения аргумента по ее буквенной модели зависит как от вида этой модели, так и от особенностей микрокалькулятора используемого типа. Если буквенная модель содержит символы элементарных функций, не вычисляемых автоматически используемым микрокалькулятором, то такую модель необходимо преобразовать к виду, содержащему только символы операторов входного языка микрокалькулятора. Вместе с тем такие модели (которые могут содержать лишь арифметические операторы) часто целесообразно преобразовать к виду, позволяющему выполнять цепные вычисления с минимальным количеством запоминаемых промежуточных результатов.

Значения степенного многочлена (27) для заданных значений аргумента на микрокалькуляторе удобно вычислять по схеме Горнера согласно формуле (33), которая для $x = \xi$ реализуется на стандартном входном языке предложением

$$|C| a_m | \times |\xi| + |a_{m-1}| (b_{m-1}) | \times |\xi| + | \dots | + \\ + |a_1| (b_1) | \times |\xi| + |a_0| = |.$$

В этом случае в процессе вычислений на индикаторе высвечиваются значения коэффициентов b_i ($b_m = a_m$) многочлена $P_1(x)$, равного частному от деления исходного многочлена на двучлен $x - \xi$.

При вычислениях конечных цепных дробей (36) можно воспользоваться схемой Горнера для деления согласно формуле

$$y = \alpha_0 + \beta_1 : (\alpha_1 + \beta_2 : (\alpha_2 + \dots + \beta_{m-1} : (\alpha_{m-1} + \\ + \beta_m : \alpha_m \dots))), \quad (116a)$$

где α_0 — первое слагаемое цепной дроби, а β_k и α_k — элементы звеньев.

В том случае, когда входной язык используемого микрокалькулятора содержит оператор $1/x$, значение цепной дроби вычисляется по формуле (116a) вводом предложения

$$|C| \alpha_m | 1/x | \times | \beta_m | + | \alpha_{m-1} | 1/x | \times | \beta_{m-1} | + | \dots | + \\ + | \alpha_1 | 1/x | \times | \beta_1 | + | \alpha_0 | = |.$$

При отсутствии оператора $1/x$ и дополнительных регистров памяти вычисление цепной дроби разбивается на ряд промежуточных вычислений согласно рекуррентной формуле

$$y_i = \beta_{i-1} : y_{i-1} + \alpha_i; \quad i = m, m-1, \dots, 1; \quad y_m = \alpha_m, \quad (116b)$$

где m — число звеньев цепной дроби.

Для извлечения квадратного корня на микрокалькуляторе, не обеспечивающем автоматического выполнения этой операции, целесообразно воспользоваться итерационной формулой (42a), рациональная форма записи которой зависит от синтаксиса входного языка.

ка микрокалькулятора используемого типа. Например, извлечение корня на микрокалькуляторе типа «Электроника Б3-24», имеющего дополнительный регистр памяти, можно выполнять вводом предложений (при $x > 1$)

$$|y_i| \cdot \Pi + |x| \div |\text{ИП}| + |\text{ИП}| \div |2| = |(y_{i+1})|.$$

В качестве начального приближения можно взять любое отличное от нуля положительное число (например, $y_0 = 1$), но для ускорения вычислений целесообразно выбирать ближайший к x квадрат целого числа. Вычисления повторяются (с вводом в каждом цикле заданного значения x) до совпадений двух вычисленных значений y_i с требуемым числом значащих цифр. Например, при вычислении $\sqrt{12,5}$ для $y_0 = 6,25$ последовательно получим $y_1 = 4,125$; $y_2 = 3,5776515$; $y_3 = 3,5357818$; $y_4 = 3,5355339$; $y_5 = 3,5355339$ с максимально достижимой точностью для 8-разрядного микрокалькулятора. Если $x < 1$, то его умножают на $a^2 > 1/x$ и результат извлечения корня из a^2x делят на a .

Для вычисления других элементарных и специальных функций, операторы которых отсутствуют во входном языке используемого микрокалькулятора, также необходимо использовать наиболее просто реализуемые разложения, обеспечивающие требуемую точность.

Вычисление членов ряда и их суммы при распространенных разложениях в степенной ряд достаточно просто реализуется на микрокалькуляторе любого типа. Достижимая точность в этом случае зависит от вида ряда, числа суммируемых членов и особенностей используемого микрокалькулятора. Для грубой оценки предельной точности результата почлененного суммирования ряда при достижении для остаточной погрешности относительного машинного нуля (машинного нуля мантиссы) можно рекомендовать следующие правила:

1. Для монотонно сходящихся рядов при вычислениях на микрокалькуляторе с естественным представлением чисел операционная погрешность

$$\varepsilon_a \approx a_q \cdot 10^{q-p} + m \cdot 10^{1-p},$$

где a_q — первая значащая цифра результата; q — его порядок; p — разрядность калькулятора; m — число членов суммируемого ряда (до номера члена, обращающегося в машинный нуль).

2. Для монотонно сходящихся рядов при вычислениях на микрокалькуляторе со стандартной формой представления чисел операционная погрешность

$$\varepsilon_a \approx a_q \cdot 10^{q-p}.$$

3. Если члены знакопеременного ряда вначале возрастают по модулю и начинают убывать лишь с s -го члена, то операционная погрешность в основном определяется погрешностью наибольших по-

модулю членов ряда, для каждого из которых ее можно принять равной единице младшего разряда. Оценивая число членов с наибольшим порядком по первой значащей цифре максимального по величине члена, можно использовать следующую приближенную формулу для оценки операционной погрешности при почленном суммировании на микрокалькуляторе любого типа:

$$\varepsilon_a \approx a_s \cdot 10^{b-p} \approx u_s \cdot 10^{1-p},$$

где b — порядок максимального по модулю члена ряда.

Так, для ряда (45а) при $x < 0$ максимальным будет член u_s с номером $s \approx x$, для оценки которого при $|x| \gg 1$ применима формула Стирлинга

$$k! \approx \left(\frac{k}{e}\right)^k \sqrt{2\pi k} \left(1 + \frac{1}{12k} + \frac{1}{288k^2} + \dots\right) \approx \left(\frac{k}{e}\right)^k \sqrt{2\pi k}.$$

Следовательно,

$$u_s \approx e^{|x|}/\sqrt{2\pi|x|} \text{ и } \varepsilon_a \approx 10^{1-p} e^{|x|}/\sqrt{2\pi|x|}.$$

Для рядов (48а) и (48б) при $|x| > 1$ номер максимального члена $s \approx (-1 + \sqrt{1 + 4x^2})/4 \approx x/2$ и

$$u_s \approx e^{|x|}/\sqrt{2\pi|x|}; \quad \varepsilon_a \approx 10^{1-p} e^{|x|}/\sqrt{2\pi|x|}.$$

Для разложения интегрального синуса в ряд (54) из уравнения $x^2(2k-1) = 2k(2k+1)^2$ следует, что максимальен член с номером $s \approx (x-1)/2$ и при $|x| \gg 1$ номер $s \approx x/2$, откуда

$$u_s \approx e^{|x|}/\sqrt{2\pi|x|^3} \text{ и } \varepsilon_a \approx 10^{1-p} e^{|x|}/\sqrt{2\pi|x|^3}.$$

Для разложения интеграла вероятности в ряд (55а) из уравнения $x^2(2k-1) = k(2k+1)$ находим

$$s \approx \frac{x^2-2}{2}; \quad u_s \approx \frac{x\sqrt{2}}{\pi(x^2-1)\sqrt{x^2-2}} \left(\frac{x^2e}{x^2-2}\right)^{\frac{(x^2-2)}{2}},$$

при $x \gg 1$ получим

$$s \approx x^2/2; \quad u_s \approx \frac{\sqrt{2}}{\pi x^2} e^{x^2/2}; \quad \varepsilon_a \approx \frac{10^{1-p}}{\pi x^2} \sqrt{2} e^{x^2/2}.$$

Разложению функций Бесселя в ряд (53а) соответствует максимальный член с номером $s = (\sqrt{x^2 + n^2} - n)/2$ и величиной

$$u_s \approx \frac{1}{n! \pi x} e^{\sqrt{x^2+n^2}} \left(\frac{\sqrt{x^2+n^2}-n}{\sqrt{x^2+n^2}+n}\right)^{n/2}.$$

При $x > n$ значение $u_s \approx e^x/n! \pi x$, но если $x < n$, то

$$u_s \approx (xe^2/2n^2)^n/\sqrt{\pi^3 x^2 n},$$

что практически при любом значении $x < n$ обеспечивает незначительную погрешность.

Оценим для примера достижимую точность вычисления интеграла вероятности (55а) почленным суммированием степенного ряда на 8-разрядном микрокалькуляторе с естественным представлением чисел при $x = 4$. По приведенным выше оценкам максимальным будет член ряда с номером $s \approx (x^2 - 2)/2 \approx 7$ и величиной $u_s = \sqrt{2/\pi} \cdot 4 \cdot 8^7/15 \cdot 7! \approx 88,3$. Следовательно, даже при отсутствии остаточной погрешности погрешность результата $\epsilon_a \approx 88,3 \cdot 10^{-7} \approx 9 \cdot 10^{-6}$. Проверка сложением 30 членов ряда дает $\Phi(4) = 0,9999404$ при точном значении с шестью верными цифрами $\Phi(4) = 0,999937$. Следовательно, истинная погрешность $\epsilon_a = 3 \cdot 10^{-6}$, что хорошо согласуется с оценкой предельной погрешности.

В большинстве инженерных расчетов достаточна точность вычисления элементарных функций с 3—4 верными знаками. Это позволяет ограничивать степенные разложения некоторых функций небольшим числом членов. Точность вычислений при небольшом числе членов ряда можно повысить экономизацией с использованием формул (40), но коэффициенты экономизированного ряда, как правило, имеют большое число значащих цифр, ввод которых в микрокалькулятор связан со значительными затратами времени. Поэтому при составлении расчетных формул для многократных вычислений с не очень высокой точностью целесообразно ограничиться коррекцией коэффициентов одного-двух членов ряда, уменьшающей остаточную погрешность.

Так, показательная функция аппроксимируется скорректированным степенным многочленом

$$e^x \approx 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{101,52}$$

с погрешностью $\epsilon_a \leq 10^{-4}$ в интервале $0 \leq x \leq 1$. Вычисления этого усеченного ряда на микрокалькуляторе удобно выполнять по формуле

$$e^x \approx \left(\left(\left(\left(\frac{x}{4,23} + 1 \right) \frac{x}{4} + 1 \right) \frac{x}{3} + 1 \right) \frac{x}{2} + 1 \right) x + 1.$$

При вычислениях тригонометрических функций их аргументы для повышения точности заключают в интервал $-\pi/4 \leq x \leq \pi/4$ (или $\pi/4 \leq x \leq \pi/2$). С этой целью заданное значение аргумента, выходящее за этот интервал, записывают в виде $z = x + n\pi/2$ и функцию аргумента заменяют такой же функцией аргумента x , если n четно, или сходственной, если n нечетно. Сходственные функции* дополнительных углов равны между собой, что позволяет заключать аргумент в интервал $[-\pi/4; \pi/4]$, используя равенства $\sin x = (\pi/2 - x)$; $\operatorname{tg} x = \operatorname{ctg}(\pi/2 - x) = \operatorname{ctg} 1/x$ и т. п.

* Сходственными называют функции $\sin x$ и $\cos x$, $\sec x$ и $\csc x$, $\operatorname{tg} x$ и $\operatorname{ctg} x$.

В некоторых микрокалькуляторах аргумент тригонометрических функций должен быть задан только в радианах или градусах. Перевод градусов α в радианы ρ реализуется формулой

$$\rho(\alpha) = \frac{\pi\alpha}{180} = \left(\left(\frac{\alpha''}{60} + \alpha' \right) \frac{1}{60} + \alpha^\circ \right) \frac{\pi}{180}, \quad (117)$$

описываемой на стандартном входном языке предложением

$$|C|\alpha''| \div |60| + |\alpha'| \div |60| + |\alpha^\circ| \times |\pi| \div |180| = |(\rho)|.$$

Перевод радиан в градусы описывается формулой

$$\alpha(\rho) = \frac{180\rho}{\pi}; \quad \alpha^\circ = E(\alpha); \quad \alpha' = E(60(\alpha - \alpha^\circ)); \quad \alpha'' = 60(\alpha_{\min} - \alpha'), \quad (118)$$

где $E_k(x)$ — целая часть десятичного представления числа x . Эта формула реализуется предложением

$$|C|180| \div |\pi| \times |\rho|(\alpha) - |\alpha^\circ| \times |60|(\alpha_{\min}) - |\alpha'| \times |60| = |(\alpha''),$$

где в индикаторный регистр вводится исходный операнд ρ , а α° и α'' определяются непосредственно по индикатору как целые части (слева от запятой) градусов α и минут α_{\min} .

Таблица 13

ρ	α	ρ	α
1	$57,29577 = 57^{\circ}17'44,8008''$	$4,848136 \cdot 10^{-6}$	$1''$
0,1	$5,729577 = 5^{\circ}43'46,4801''$	6,283185	360°
0,01	$0,5729577 = 34'22,64797''$	3,141593	180°
0,001	$5,729577 \cdot 10^{-2} = 3'26,2648''$	1,570796	90°
0,0001	$5,729577 \cdot 10^{-3} = 20,62647''$	1,553343	89°
0,00001	$5,729577 \cdot 10^{-4} = 2,062648''$	1,178097	$67,5^\circ$
0,000001	$5,729577 \cdot 10^{-5} = 0,2062648''$	1,047198	60°
0,0000001	$5,729577 \cdot 10^{-6} = 0,02062648''$	0,7853982	45°
0,01745329	$1'$	0,5235988	30°
$2,908882 \cdot 10^{-4}$	$1'$	0,3926991	$22,5^\circ$

Для справок в табл. 13 приведены соотношения между некоторыми значениями аргументов тригонометрических функций в радианах и градусах. Из табличных данных, в частности, следует, что погрешность в единице шестого разряда после запятой десятичного представления радиан составляет ошибку более двух секунд. Задаваясь допустимой точностью аргумента в градусной мере, по таблице можно оценить требуемое число значащих цифр аргумента в радианах.

При инженерных расчетах тригонометрические функции можно вычислять с помощью следующих скорректированных степенных разложений:

$$\sin x = x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{120d} = - \left(\left(\left(\frac{x^2}{d} - 1 \right) \frac{x^2}{20} + 1 \right) \frac{x^2}{6} - 1 \right) x, \quad (119a)$$

обеспечивающих для 8-разрядного микрокалькулятора при $d = 42,25$ погрешность $\epsilon_a \leq 10^{-7}$ в интервале $0 \leq x \leq \pi/4$ и при $d = 43,36$ погрешность $\epsilon_a \leq 1,1 \cdot 10^{-5}$ в интервале $0 \leq x \leq \pi/2$;

$$\cos x = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{24d} = - \left(\left(\left(\frac{x^2}{d} - 1 \right) \frac{x^2}{12} + 1 \right) \frac{x^2}{2} - 1 \right), \quad (119b)$$

обеспечивающих при $d = 30,31$ и $0 \leq x \leq \pi/4$ погрешность $\epsilon_a \leq 4 \cdot 10^{-7}$, а при $d = 31,24$ и $0 \leq x \leq \pi/2$ погрешность $\epsilon_a \leq 7 \cdot 10^{-5}$;

$$\operatorname{tg} x = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{2x^7}{15d} = \left(\left(\left(\frac{x^2}{d} + 1 \right) \frac{2x^2}{5} + 1 \right) \frac{x^2}{3} + 1 \right) x, \quad (119b)$$

обеспечивающих $\epsilon_a \leq 3 \cdot 10^{-4}$ при $d = 1,89$ и $0 \leq x \leq \pi/4$.

Логарифмическую функцию целесообразно вычислять по формуле (46в), в которой для выбора числа членов ряда, заключенного в скобки, можно руководствоваться следующими оценками: максимальная остаточная погрешность (при $W = 0,3$) равна 0,00026 при трех членах ряда; 0,000023 при четырех; 0,0000021 при пяти и 0,00000019 при шести. Если используется восьмиразрядный калькулятор, не имеющий оператора $1/x$, то сумму членов в скобках формулы (46в) можно вычислять по формуле

$$y = W(1 + W^2(0,3333333 + W^2(0,2 + W^2(0,1428571 + W^2(0,1111111 + 0,0909091W^2)))), \quad (120)$$

уменьшая число внутренних скобок, если требуемая точность меньше достижимой.

Например, для вычисления $\lg 18,2$ с точностью до четырех верных знаков согласно формуле (47) следует взять $M = 0,43429$ и вычислить $\ln 18,2$ с точностью до пяти верных знаков. Представим $18,2 = 2^5 \cdot 0,56875$. Тогда $W = 0,2749$ и согласно приведенным выше оценкам достаточно взять четыре члена ряда в скобках формулы (46в). Вычисляя сумму четырех членов этого ряда, получим $y = 0,282156$ и по формуле (46в) $\ln x = 5 \cdot 0,693147 - 2 \times 0,282156$. Искомый результат $\lg = 0,43429 \cdot 2,901423 = 1,260058 \approx 1,26$ с требуемой точностью.

Разложения в степенные ряды с медленной сходимостью целесообразно преобразовывать в дробно-рациональные выражения или в цепные дроби в соответствии с формулами (35), (36) и (37), так как преобразованные выражения обычно сходятся быстрее. Кроме того, преобразование степенного многочлена в цепную дробь позволяет уменьшить число операций и, следовательно, операционную погрешность.

Например, для усеченного ряда в соответствии с формулой (35)

$$\operatorname{tg} x \approx x \left(1 + \frac{x^2}{3} + \frac{2x^4}{15}\right) = x \frac{b_0 + b_1 x^2}{1 + c_1 x^2};$$

$$1 + \left(c_1 + \frac{1}{3}\right)x^2 + \left(\frac{c_1}{3} + \frac{2}{15}\right)x^4 = b_0 + b_1 x^2.$$

Составив систему уравнений

$$b_0 = 1; \quad b_1 = c_1 + 1/3; \quad 0 = c_1/3 + 2/15,$$

откуда $b_0 = 1; \quad c_1 = -2/5; \quad b_1 = -1/15$ и, следовательно,

$$\operatorname{tg} x \approx \left(\frac{1 - x^2/15}{1 - 2x^2/5}\right)x = \frac{x}{1 - \frac{x^2}{1 - \frac{x^2}{3 - \frac{x^2}{5}}}}.$$

После коррекции последнего коэффициента получаем удобную расчетную формулу

$$\operatorname{tg} x \approx \frac{x}{1 - \frac{x^2}{1 - \frac{x^2}{3 - \frac{x^2}{4,917}}}} \quad (121a)$$

с погрешностью $\epsilon_a \leq 1,7 \cdot 10^{-5}$ в интервале $-\pi/4 \leq x \leq \pi/4$, что значительно лучше, чем для степенного ряда даже с большим числом членов.

Экспоненциальную функцию можно вычислять по цепной дроби

$$e^x \approx 1 + \frac{x}{1 - \frac{x}{2 + \frac{x}{3 - \frac{x}{2,21}}}}, \quad (121b)$$

обеспечивающей погрешность $\epsilon_a \leq 2 \cdot 10^{-4}$ в интервале $0 \leq x \leq 1$ или по цепной дроби

$$e^x \approx \frac{1}{1 - \frac{2x}{2 + x + \frac{x^2}{6 + \frac{x^2}{10,07}}}}, \quad (121b)$$

позволяющей на 8-разрядном микрокалькуляторе получить в интервале $0 \leq x \leq 1$ погрешность $\epsilon_a \leq 2,5 \cdot 10^{-6}$.

Разложение $\lg(1 + x) \approx$

$$\frac{0,4343x}{1 + \frac{x}{2 + \frac{x}{3 + \frac{x}{1 + \frac{x}{6,375 + 0,68x}}}}} \quad (121g)$$

обеспечивает в интервале $1 \leqslant 1+x \leqslant 10$ погрешность $\epsilon_a \leqslant 2 \times 10^{-4}$, а цепная дробь

$$\operatorname{arctg} x \approx \frac{x}{1 + \frac{x^2}{3 + \frac{4x^2}{5 + 1,08x^2}}} \quad (121\text{d})$$

в интервале $0 \leqslant x \leqslant 1$ имеет погрешность в радианах $\epsilon_a \leqslant 10^{-4}$.

При вычислениях цепных дробей с неизвестным числом звеньев, требуемых для обеспечения заданной точности, используют формулу (38) для k -х подходящих дробей, продолжая их вычисления до тех пор, пока очередные значения k -х подходящих дробей не будут совпадать до требуемого числа значащих цифр. Для вычисления k -х подходящих дробей используют вычислительный бланк (табл. 14), расположение элементов цепной дроби в котором удобно для запоминания при использовании формулы (38).

Таблица 14

k	β_k	α_k	a_k	b_k	a_k/b_k
-1			1	0	—
0	1	α_0	α_0	1	
1	β_1	α_1			
2	β_2	α_2			
3	β_3	α_3			
...

При ориентировочных расчетах целесообразно использовать еще более простые формулы с небольшой, но достаточной по условиям задачи точностью. Например, формула

$$\sin x \approx x(1 - x^2/d)^2 \quad (122\text{a})$$

при $d = 12,18$ и $0 \leqslant x \leqslant \pi/2$ обеспечивает погрешность $\epsilon_a \leqslant 1,2 \times 10^{-3}$, а при $d = 12,052$ и $0 \leqslant x \leqslant \pi/4$ — погрешность $\epsilon_a \leqslant 5 \times 10^{-5}$. Формула

$$\cos x \approx 1 - (x^2/2)(1 - x^2/d)^2 \quad (122\text{b})$$

при $d = 24,67$ и $0 \leqslant x \leqslant \pi/2$ дает ошибку $\epsilon_a \leqslant 8 \cdot 10^{-4}$, а при $d = 24,17$ и $0 \leqslant x \leqslant \pi/4$ погрешность $\epsilon_a \leqslant 1,4 \cdot 10^{-5}$.

Если используемый микрокалькулятор имеет оператор $\sqrt[3]{}$, то в качестве аппроксимирующих формул можно использовать и иррациональные алгебраические выражения. Так, для функции $\operatorname{arc} \operatorname{tg} x$ имеющей особое значение при инженерных расчетах, приближение

$$\operatorname{arc} \operatorname{tg} x \approx 57x/\sqrt{1 + 0,607x^2} \quad (123\text{a})$$

обеспечивает в интервале $0 \leq x \leq 1$ вычисление $\arctg x$ в градусах с погрешностью менее $0,05^\circ$, а приближение

$$\arctg x \approx 80,67x / \sqrt{1 + \sqrt{1 + x^2} + 0,805x^2} \quad (123б)$$

позволяет вычислять в интервале $0 \leq x < \infty$ значение $\arctg x$ с ошибкой менее $0,09^\circ$, что достаточно для большинства инженерных приложений.

При использовании микрокалькулятора, входной язык которого не содержит оператора $\sqrt[3]{}$, пригодно приближение

$$\arctg x \approx 57x(1 - 0,3x^2)(1 - 0,3x^2), \quad (123в)$$

обеспечивающее погрешность менее $0,15^\circ$ в интервале $0 \leq x \leq 1$.

По значению функции $\varphi = \arctg x$ несложно определить и значения остальных обратных тригонометрических функций:

$$\begin{aligned} \arcsin x &= \arctg(x/\sqrt{1-x^2}); & \arccos x &= \arctg(1/\sqrt{1-x^2}); \\ & \arcctg x = \arctg(1/x). \end{aligned}$$

Например,

$$\arcsin x \approx 80,67x / \sqrt{1 + \sqrt{1 - x^2} - 0,195x^2}$$

с погрешностью менее $0,09^\circ$ в интервале $0 \leq x \leq 1$.

Для чисел с ограниченной разрядностью законы ассоциативности, коммутативности и дистрибутивности нарушаются, в связи с чем при изменении последовательности вычислений по приведенным формулам значения погрешностей также могут изменяться.

5. РЕШЕНИЕ НЕЛИНЕЙНЫХ И ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Простейший способ определения вещественных корней нелинейных алгебраических и трансцендентных уравнений при использовании непрограммируемых микрокалькуляторов заключается в использовании метода проб с контролем по индикатору текущих значений функции для пробных значений аргумента. Для отделения корней выбирают начальный интервал аргумента, в котором значения функции заведомо не изменяются по знаку, после чего увеличивают интервал до изменения знака функции на одной из его границ. Для уточнения корней интервал нахождения корня делят на две части, выбирая в простейшем случае точку деления по средине интервала или ближе к той из его границ, для которой значение функции меньше по модулю. В качестве уточненного интервала нахождения корня выбирают ту часть предыдущего интервала, на границах которой значения функции различны по знаку. Уточнение корня оканчивают, когда ширина интервала с различными знаками функции на границах не превышает заданной погрешности, обычно

определя емой числом совпадающих значащих цифр в значениях аргумента на границах интервала.

Алгебраические многочлены нечетной степени имеют по крайней мере один вещественный корень. Поэтому после выделения всех вещественных корней такого многочлена оставшийся множитель степени m имеет $m/2$ пар комплексно-сопряженных корней.

Алгебраические многочлены степени $m = 4$ разбивают на множители второй степени согласно формуле (62) методом последовательных приближений (итераций). Для этого выбирают некоторые начальные значения $p^{(0)}$ и $q^{(0)}$ коэффициентов выделяемого квадратичного множителя (часто принимают $p^{(0)} = a_3/a_4$; $q^{(0)} = a_2/a_4$ или $p^{(0)} = -a_1/a_2$; $q^{(0)} = a_0/a_2$) и уточняют текущие значения этих коэффициентов

$$p^{(k+1)} = p^{(k)} + \Delta p^{(k)}; \quad q^{(k+1)} = q^{(k)} + \Delta q^{(k)}.$$

Поправки Δp и Δq находят на каждой итерации решением системы уравнений

$$c_2\Delta p + c_3\Delta q = b_1; \quad c_1\Delta p + c_2\Delta q = b_0$$

по формулам

$$\Delta p = (b_1c_2 - b_0c_3)/D; \quad \Delta q = (b_0c_2 - b_1c_1)/D; \quad D = c_2^2 - c_1c_3.$$

Коэффициенты системы уравнений вычисляют по коэффициентам a_i исходного многочлена четвертой степени, последовательно используя формулы

$$\begin{aligned} b_4 &= c_4 = a_4; \quad b_3 = a_3 - pb_4; \quad b_2 = a_2 - pb_3 - qb_4; \\ b_1 &= a_1 - pb_2 - qb_3; \quad b_0 = a_0 - pb_1 - qb_2; \quad c_3 = b_3 - pc_4; \\ c_2 &= b_2 - pc_3 - qc_4; \quad c_1 = -pc_2 - qc_3. \end{aligned}$$

Вычисления оканчиваются при достаточно малых значениях поправок Δp и Δq или коэффициентов b_1 и b_0 . Требуемое число итераций зависит от начального выбора коэффициентов p и q , но резко возрастает при близких по модулю корнях уравнений.

После выделения квадратичного множителя и деления на него исходного многочлена искомые корни определяются по формуле (61). При вычислении корней многочлена четвертой степени рассмотренным способом форма вычислительного бланка зависит от типа используемого микрокалькулятора. Если микрокалькулятор обеспечивает выполнение операций «со скобками» или имеет дополнительные регистры памяти, то количество промежуточных результатов, записываемых в бланк, соответственно уменьшается. Для удобства расчета целесообразно в начальной части бланка записать формулы (или соответствующие предложения на входном языке микрокалькулятора), по которым вычисляются записываемые в бланк результаты.

При $m = 6$ вычисление корней рассмотренным способом громоздко и в этом случае используют другие методы (например, Ло-

бачевского — Греффе) определения комплексно-сопряженных корней.

Пример 19. Вычислить корни алгебраического многочлена

$$P(x) = x^5 + 2,2x^4 + 4,45x^3 + 6,15x^2 + 4,95x + 2,7$$

с помощью простейшего микрокалькулятора.

Заданный многочлен имеет по крайней мере один вещественный отрицательный (так как все коэффициенты положительны) корень. Записав многочлен в удобной для вычислений форме

$$P(x) = (((x + 2,2)x + 4,45)x + 6,15)x + 4,95)x + 2,7,$$

выберем начальный интервал аргумента $(0; -1)$, в котором значение многочлена не изменяется по знаку ($|a_0| > 1$). При удвоении интервала значение функции изменяется по знаку и, следовательно, корень находится в интервале $(-1; -2)$.

Для уточнения корня последовательно делим этот интервал на меньшие части в точках $x = -1,5; -1,25; -1,1; -1,2$. Для последнего значения аргумента $P(-1,2) = 0$, что соответствует искомому корню $x_1 = -1,2$.

Разделив заданный многочлен на множитель $x + 1,2$ вещественного корня по схеме Горнера в соответствии с формулой (33), получаем многочлен четвертой степени

$$P_1(x) = x^4 + x^3 + 3,25x^2 + 2,25x + 2,25.$$

Приняв $p^{(0)} = a_1/a_2 \approx 0,7$; $q^{(0)} = a_0/a_2 \approx 0,7$, составляем вычислительный бланк (табл. 15), в начальный столбец которого записываем расчетные формулы (с учетом коэффициентов многочлена четвертой степени), а в последующие столбцы — результаты вычислений на каждой k -й итерации.

В результате вычислений получим (с проверкой подстановкой) $p = 1$; $q = 1$. Определяя делением оставшийся множитель, окончательно получим

$$P(x) = (x + 1,2)(x^2 + x + 1)(x^2 + 2,25)$$

или, в соответствии с формулой (61)

$$x_1 = -1,2; x_{2,3} = -0,5 \pm j1,5; x_{4,5} = \pm j0,866.$$

Для вычисления корней методом Лобачевского — Греффе составляем вычислительный бланк (табл. 16), в который записываем исходные значения

k	1	2
$b_3 = 1 - p$	0,3	-0,0445775
$b_2 = -b_3p - q + 3,25$	2,34	2,352590
b_2p	1,638	2,457464
$b_1 = -b_3q - b_2p + 2,25$	0,402	-0,1653827
b_1p	0,2814	-0,1727550
$b_0 = -b_2q - b_1p + 2,25$	0,3306	0,2019699
$c_3 = -2p + 1$	-0,4	-1,089155
$c_2 = -c_3p - q + b_2$	1,92	2,546322
c_2p	1,344	2,659831
$c_1 = -c_3q - c_2p$	-1,064	-1,631696
c_1c_3	0,4256	1,777170
$D = c_2^2 - c_1c_3$	3,2608	4,706587
b_0c_1	-0,3517584	-0,3295535
b_0c_3	-0,1608	0,1801273
$p^{(k+1)} = (c_2b_1 - b_0c_1) : D + p$	1,044577	1,025123
$q^{(k+1)} = (c_2b_0 - b_1c_3) : D + p$	0,943744	1,014971

коэффициентов заданного многочлена и результаты их квадрирования по формуле (63б). Так как после семи квадрирований разделяются только коэффициенты a_3 и a_2 , то многочлен имеет один вещественный и две пары комплексно-сопряженных корней.

По формуле (64а) вычисляем

$$|x_1| = e^{\frac{1}{2^7} \ln \frac{1,6389328 \cdot 10^{56}}{1,2005001 \cdot 10^{46}}} = e^{\frac{23,337156}{128}} = 1,1999996.$$

Так как все коэффициенты многочлена положительны, то $x_1 = -1,1999996 \approx -1,2$.

По формуле (64б) вычисляем

$$|x_{2,3}| = e^{\frac{1}{2} \frac{2}{2^7} \ln 1,2005001 \cdot 10^{46}} = 1,4999995;$$

$$|x_{4,5}| = e^{\frac{1}{2} \frac{1}{2^7} \ln \frac{1,6389344 \cdot 10^{56}}{1,6389328 \cdot 10^{56}}} = 1.$$

В соответствии с основной теоремой алгебры

$$-a_4/a_5 = x_1 + 2 \operatorname{Re} x_{2,3} + 2 \operatorname{Re} x_{4,5};$$

$$a_1/a_5 = |x_{2,3}|^2 |x_{4,5}|^2 + 2x_1 (|x_{4,5}|^2 \operatorname{Re} x_{2,3} + |x_{2,3}|^2 \operatorname{Re} x_{4,5})$$

или

$$\operatorname{Re} x_{2,3} + \operatorname{Re} x_{4,5} = -0,5000002;$$

$$\operatorname{Re} x_{2,3} + 2,249998 \operatorname{Re} x_{4,5} = -1,1250010.$$

Отсюда

$$\operatorname{Re} x_{4,5} = 1,24 \cdot 10^{-6}; \quad \operatorname{Re} x_{2,3} = -0,5000014;$$

$$\operatorname{Im} x_{2,3} = \sqrt{1,4999995^2 - 0,00000228^2} = 1,4999994;$$

$$\operatorname{Im} x_{4,5} = \sqrt{1 - 0,5000012^2} = 0,8660247.$$

Округляя полученные результаты до пяти значащих цифр, получаем

$$x_1 = -1,2; \quad x_{2,3} = -0,5 \pm j 1,5; \quad x_{4,5} = \pm j 0,866.$$

Таблица 15

3	4	5	6
-0,0251231	-0,0002798	0,0004227	0,0000106
2,260783	2,247185	2,249739	2,250041
2,317580	2,247813	2,248788	2,250017
-0,0420815	0,0024669	0,0007894	-0,0000277
-0,0431387	0,0024676	0,0007891	-0,0000277
-0,0014909	-0,0066072	-0,0001647	0,0001029
-1,050246	-1,000559	-0,9991546	-0,9999787
2,322443	2,244929	2,248632	2,250060
2,380289	2,245557	2,247682	2,250036
-1,314820	-1,241901	-1,248689	-1,250109
1,380884	1,242596	1,247633	1,250083
4,012857	3,797113	3,808716	3,812690
0,0019603	0,0082055	0,0002057	-0,0001286
0,0441959	-0,0024683	-0,0007887	0,0000280
1,000279	0,9995773	0,9999893	1,0000060
1,003094	0,9998385	0,9999483	1,0000010

Таблица 16

<i>j</i>	<i>a₅</i>	<i>a₄</i>	<i>a₃</i>	<i>a₂</i>	<i>a₁</i>	<i>a₀</i>
0	1	2,2	4,45	6,15	4,95	2,7
1	1	4,84	19,8025	37,8225	24,5025	7,29
		-8,9	-27,06	-44,055	-33,21	
	1	-4,06	2,6425	5,6475	-8,7075	7,29
2	1	16,4836	6,9828062	31,894256	75,820556	53,1441
		-5,285	45,8577	46,019137	-82,34055	
	1	11,1986	-17,415	-59,1948		
3	1	125,40864	1254,9664	350,38561	42,510321	2824,2953
		-70,851012	-419,244	461,94817	-1989,5652	
	1	54,55763	-13,039988	1190,279		
4	1	2976,5349	676806,33	4010458	3791022,7	7976643,9
		-1645,3648	-218515,61	3203615,5	-11311939	
	1	1331,1701	-3894,1098	308173,71		
5	1	1772013,8	2,0647627 · 10 ¹¹	5,6584202 · 10 ¹³	5,6564177 · 10 ¹³	6,3626847 · 10 ¹³
		-908793,22	-2,0026781 · 10 ¹⁰	6,8349574 · 10 ¹²	-1,2000457 · 10 ¹⁴	
	1	863220,60	-1,5041832 · 10 ⁷	2,1236539 · 10 ¹⁰		
6	1	7,451498 · 10 ¹¹	1,8643445 · 10 ¹¹	6,3440396 · 10 ¹³	-6,344039 · 10 ¹³	6,3626847 · 10 ¹³
		-3,728689 · 10 ¹¹	3,4757884 · 10 ²²	4,0246838 · 10 ²⁷	4,024683 · 10 ²⁷	4,0483756 · 10 ²⁷
	1	3,722809 · 10 ¹¹	-1,0952611 · 10 ²⁰	2,3654948 · 10 ²⁵	-8,0730246 · 10 ²⁷	
7	1	1,3859306 · 10 ²³	-1,2688078 · 10 ¹⁴	1,0984801 · 10 ²⁰		
		-6,9296556 · 10 ²²	3,4648278 · 10 ²²	4,0483389 · 10 ²⁷	-4,04834161 · 10 ²⁷	4,0483756 · 10 ²⁷
	1	6,92965 · 10 ²²	1,2005001 · 10 ⁴⁵	1,6389328 · 10 ⁵⁵	-1,6389323 · 10 ⁵⁵	1,6389344 · 10 ⁵⁵

В тех случаях, когда по условиям задачи достаточно оценить устойчивость алгебраического многочлена без вычисления его корней, можно использовать один из критериев (65). Распространенный критерий Рауса — Гурвица (65а) при анализе многочленов высокой степени связан с громоздкими вычислениями определителей, значения которых при плохо обусловленных матрицах (близости определителей к нулю) должны быть вычислены особенно точно. Поэтому при использовании микрокалькуляторов для оценки устойчивости многочленов высокой степени более удобен критерий Гурвица (65б).

Разложение в цепную дробь при использовании критерия Гурвица оканчивается преждевременно (до вычисления положительных значений всех m коэффициентов цепной дроби), если четная и нечетная части анализируемого многочлена имеют общий множитель в виде четных многочленов $x^2 + a$, $x^4 + b_1x^2 + b_0$ или их произведения. Корни таких четных множителей расположены симметрично относительно оси $j\omega$ (в частности, на этой оси) и, следовательно, алгебраический многочлен, содержащий подобный множитель, неустойчив. Поэтому преждевременное окончание разложения в цепную дробь до вычисления всех m коэффициентов также свидетельствует о неустойчивости, что учтено в формуле (65б).

Обобщенный критерий Найквиста (65в) удобно использовать в тех случаях, когда по условиям задачи линейное дифференциальное уравнение отображено в частотную область алгебраическим многочленом. Однако этот критерий в равной мере применим и для любых алгебраических многочленов.

Пример 20. Оценить устойчивость многочлена, заданного в предыдущем примере.

Для сравнения используем все критерии (65). В соответствии с формулами (65а) определяем

$$a_5 = 1 > 0; a_4 = 2,2 > 0; a_0 = -2,7 > 0;$$

$$\begin{vmatrix} 2,2 & 1 \\ 6,15 & 4,45 \end{vmatrix} = 3,64 > 0; \begin{vmatrix} 2,2 & 1 & 0 \\ 6,15 & 4,45 & 2,2 \\ 2,7 & 4,95 & 6,15 \end{vmatrix} = 4,368 > 0;$$

$$\begin{vmatrix} 2,2 & 1 & 0 & 0 \\ 6,15 & 4,45 & 2,2 & 1 \\ 2,7 & 4,95 & 6,15 & 4,45 \\ 0 & 0 & 2,7 & 4,95 \end{vmatrix} = \det \begin{matrix} 1 & 1 & 2 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 1 & 2 & 3 & 4 \\ 4 & 3 & 4 & 1 & 2 & 3 & 4 \end{matrix}_{\text{mod } 2} =$$

$$\begin{aligned} &= \det \begin{matrix} 1 & 2 \\ 2 & 3 & 4 & 1 & 3 & 4 \\ 3 & 3 & 4 & 2 & 2 & 3 & 4 & 1 & 1 \\ 4 & 4 & 3 & 4 & 3 & 4 & 3 & 4 & 3 \end{matrix} = a_{11} [a_{22}(a_{33}a_{44} - a_{34}a_{43}) + a_{32}(a_{24}a_{43} - a_{23}a_{44})] + \\ &\quad + a_{12} [a_{21}(a_{34}a_{43} - a_{33}a_{44}) + a_{31}(a_{23}a_{44} - a_{24}a_{43})] = 0. \end{aligned}$$

Таким образом, в соответствии с критерием Рауса — Гурвица анализируемый многочлен неустойчив и имеет корень (известный из предыдущего примера) на мнимой оси.

Для оценки устойчивости по критерию Гурвица представим заданный многочлен суммой многочленов с четными и нечетными степенями

$$P(x) = N(x) + M(x) = (x^5 + 4,45x^3 + 4,95x) + (2,2x^4 + 6,15x^2 + 2,7)$$

и составим отношение

$$\frac{N(x)}{M(x)} = \frac{x^5 + 4,45x^3 + 4,95x}{2,2x^4 + 6,15x^2 + 2,7}.$$

Разложим эту дробно-рациональную функцию в цепную дробь

$$\begin{aligned} & 2,2x^4 + 6,15x^2 + 2,7 \Big| x^5 + 4,45x^3 + 4,95x \left(\frac{x}{2,2} \right) \\ & \quad - x^5 + \frac{6,15x^3}{2,2} + \frac{2,7x}{2,2} \\ & \quad \underline{\frac{3,64x^3}{2,2} + \frac{8,19x}{2,2}} \Big| 2,2x^4 + 6,15x^2 + 2,7 \left(\frac{4,84x}{3,64} \right) \\ & \quad \quad \quad \underline{\frac{2,2x^4 + 4,95x^2}{1,2x^2 + 2,7}} \\ & 1,2x^2 + 2,7 \Big| \frac{3,64x^3}{2,2} + \frac{8,19x}{2,2} \left(\frac{3,64x}{2,64} \right) \\ & \quad \quad \quad \underline{\frac{3,64x^3}{2,2} + \frac{9,828x}{2,64}} \\ & \quad \quad \quad \underline{0} \end{aligned}$$

Так как разложение в цепную дробь закончилось делением без остатка до вычисления всех коэффициентов дроби, то анализируемый многочлен неустойчив, а его четная и нечетная части имеют общий четный множитель.

Для оценки устойчивости по обобщенному критерию Найквиста представим заданный многочлен в области $x = j\omega$ как

$$\begin{aligned} P(j\omega) &= \operatorname{Re} P(\omega) + j \operatorname{Im} P(\omega) = \\ &= (2,2\omega^4 - 6,15\omega^2 + 2,7) + \\ &+ j\omega(\omega^4 - 4,45\omega^2 + 4,95). \end{aligned}$$

Задаваясь значениями $\omega_i + i = \omega_i + \Delta\omega$ при $\omega_0 = 0$ и $\Delta\omega = 0,1$, строим на комплексной плоскости многочлена $P(j\omega)$ его частотный годограф (рис. 12). Пересечение годографом критической точки в начале координат свидетельствует о неустойчивости заданного многочлена, один из корней которого расположен на оси $j\omega$.

При решении инженерных задач с нелинейными уравнениями

целесообразно вычислять лишь те их корни, которые имеют физический смысл или представляют интерес по условиям задачи. Это особенно существенно при решении трансцендентных уравнений, связанном при использовании микрокалькуляторов, входные языки которых не содержат требуемых операторов автоматического вычисления элементарных функций, с большими затратами времени.

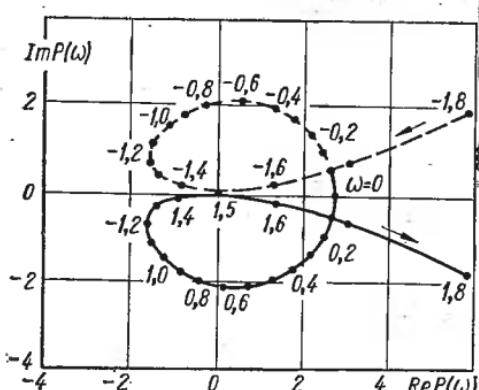


Рис. 12. Частотный годограф многочлена $P(x) = x^5 + 2,2x^4 + 4,45x^3 + 6,15x^2 + 4,95x + 2,7$

Поэтому при решении таких уравнений грубую оценку интервалов нахождения искомых корней производят с учетом физической постановки задачи.

Вычисление вещественных корней трансцендентных уравнений при помощи микрокалькуляторов, входные языки которых содержат требуемые операторы вычисления элементарных функций, часто удобно производить методом деления интервала нахождения корня. В этом случае может потребоваться большее число итераций, чем при использовании более сложных методов, но общие затраты времени (в связи с простотой вычислительной процедуры в методе деления) могут оказаться меньшими.

Пример 21. Определить с точностью до трех значащих цифр значение тока через диод с уравнением статической характеристики $I = 10^{-4}(e^{40U} - 1)$, который включен последовательно с идеальным источником напряжения $E = 4$ В и резистором с сопротивлением $R = 10$ Ом.

В соответствии с законом Кирхгофа $E = RI + U = 10^{-4}R(e^{40U} - 1) + U$, откуда $f(U) \equiv 10^{-3}(e^{40U} - 1) + U - 4 = 0$.

Выделим грубо интервал корня, задаваясь значениями $U = 0; 0,1; 0,2; 0,3$ и вычисляя соответственно $f(U) = -4; -3,8; -0,82; 159,0$. Корень лежит в интервале $[0,2; 0,3]$, на границах которого значения функции противоположны по знаку.

Производные $f'(U) = 4 \cdot 10^{-2}e^{40U} + 1$ и $f''(U) = 1,6e^{40U}$ в найденном интервале изменяются монотонно и на границах интервала $f'(0,2) = 120,24$; $f'(0,3) = 6511$; $f''(0,2) = 4769,5$; $f''(0,3) = 260\,407$. Для выполнения условия сходимости метода хорд дополнительно сокращаем интервал нахождения корня до отрезка $[0,2; 0,21]$, на верхней границе которого $f'(0,21) = 178,9$ и $f(0,21) = 0,65607$.

Так как знаки функции и ее второй производной совпадают на верхней границе интервала, принимаем ее фиксированной и производим вычисления методом хорд (табл. 17) в соответствии с первой формулой (56). Значение $|x_2 - x_1| = 0,00045 < \epsilon = 0,0005$ и, следовательно, искомый корень $U = 0,206$ определен с заданной точностью.

Для вычисления корня методом касательных принимаем $U_0 = 0,21$ и записываем результаты вычислений в соответствии с формулой (57) в бланк, приведенный в табл. 17. Вычисления заканчиваем по достижении заданной точности.

Метод итераций согласно формуле (55) можно использовать после представления заданного уравнения в форме

$$U = \ln [1000(4 - U) + 1]/40.$$

Результаты вычислений методом итераций и методом проб для исходного интервала $(0,2; 0,21)$ приведены в табл. 17.

Из сравнения результатов вычисления вещественного корня трансцендентного уравнения различными методами в рассмотренном примере видно, что, как и при вычислении вещественного корня алгебраических уравнений, достаточно удобен метод проб с выделением и уточнением корня соответственно расширением и сужением интервала аргумента. Особенно удобен метод проб при вычислениях на микрокалькуляторах с дополнительным регистром памяти для занесения предыдущих значений функции и их сравнения с текущими. Однако в общем случае метод проб с делением интервала со-

Метод хорд

Итерация	0	1	2
$\frac{U}{f(U)}$ $b - U$ $f(b) - f(U)$	0,2 -0,82 0,01 1,4711	0,20555 -0,073512 0,00445 0,729582	0,205998

Метод касательных

Итерация	0	1	2
$\frac{U}{f'(U)}$	0,21 0,65607 177,9	0,2063328 0,0456568 154,61288	0,2060376

Метод итераций

Итерация	0	1	2	3	4
U	0,21	0,2060096	0,2060359	0,2060357	0,2060357

Метод проб

Итерация	0	1	2	3	4
$\frac{U}{f(U)}$	0,2 -0,82	0,21 0,65607	0,205 -0,15505	0,206 -0,00546	0,2061 0,00982

дится медленнее других методов решения трансцендентных уравнений.

При приближенном решении обыкновенных дифференциальных уравнений с помощью микрокалькулятора иногда целесообразно использовать аналитические методы, в частности, ранее рассмотренные методы последовательного приближения и последовательного дифференцирования с составлением степенного ряда Тейлора.

Пример 22. Решить приближенными аналитическими методами дифференциальное уравнение

$$y' = y - 4x + 3$$

с начальным условием $x_0 = 0$, $y_0 = 3$ в интервале $0 \leq x \leq 0,5$.

Используя метод последовательных приближений, составляем интегральное уравнение

$$y(x) = 3 + \int_{x_0}^x (y - 4x + 3) dx$$

и последовательно определяем

$$y_1(x) = 3 + \int_{x_0}^x (6 - 4x) dx = 3 + 6x - 2x^2;$$

$$y_2(x) = 3 + \int_{x_0}^x (6 + 2x - 2x^2) dx = 3 + 6x + x^2 - \frac{2}{3}x^3;$$

$$y_3(x) = 3 + \int_{x_0}^x \left(6 + 2x + x^2 - \frac{2}{3}x^3\right) dx = 3 + 6x + x^2 + \frac{x^3}{3} - \frac{x^4}{6};$$

$$y_4(x) = 3 + \int_{x_0}^x \left(6 + 2x + x^2 + \frac{x^3}{3} - \frac{x^4}{6}\right) dx = 3 + 6x + x^2 + \frac{x^3}{3} + \frac{x^4}{12} - \frac{x^5}{30}.$$

Применяя метод степенного ряда (последовательного дифференцирования), находим $y'' = y' - 4 = y + 4x - 1 = \dots = y^{(k)}$ и с учетом начальных условий $y'_0 = 6; y''_0 = \dots = y^{(n-1)}_0 = 2$ получаем решение для точки $x_0 = 0$ в виде ряда

$$y = y_0 + y'_0(x - x_0) + \frac{y''_0}{2!}(x - x_0)^2 + \dots = 3 + 6x + x^2 + \frac{x^3}{3} + \frac{x^4}{12} + \dots$$

Ограничавшись пятью членами ряда, совпадающими для обоих методов, составляем расчетную формулу

$$y = \left(\left(\left(\frac{x}{4} + 1 \right) \frac{x}{3} + 1 \right) x + 6 \right) x + 3,$$

реализуемую вводом в микрокалькулятор одного предложения. Результаты вычислений по этой формуле записываем в вычислительный бланк (табл. 18), выбрав шаг $h = 0,1$.

Точное решение заданного уравнения описывается формулой $y = 2e^x + 4x + 1$. Сравнивая точные значения и результаты вычислений по составленному усеченному ряду, убеждаемся в малой погрешности последних.

Несмотря на высокую точность результатов решения дифференциального уравнения, заданного в рассмотренном примере, в общем случае аналитические методы отличаются рядом недостатков и основными являются численные методы. Выбор численного метода приближенного решения дифференциального уравнения зависит от требуемой точности результатов. Для приближенных вычислений с низкой точностью целесообразно использовать формулу (69а) метода Эйлера с относительно большим шагом.

Более высокой точностью (при большем числе операций) отличается модифицированный метод Эйлера, описываемый формулой (69б). При использовании этого метода сначала вычисляют значение функции $y_{i+0,5} = y_i + hy'_i/2$ в точке $x_{i+0,5} = x_i + h/2$, затем значение правой части дифференциального уравнения (66) в этой точке по формуле $y'_{i+0,5} = f(x_{i+0,5}, y_{i+0,5})$ и, наконец, искомое значение $y_{i+1} = y_i + hy'_{i+0,5}$.

Таблица 18

Метод	<i>i</i>	0	1	2	3	4	5
	<i>x_i</i>	0	0,1	0,2	0,3	0,4	0,5
Точно	<i>y_i</i>	3	3,610341	4,242805	4,899717	5,583649	6,297442
Ряд Тейлора	<i>y_i</i>	3	3,610341	4,2428	4,899675	5,583466	6,296875
Эйлера	<i>y_i</i> <i>y'_i</i>	3 6	3,6 6,2	4,22 6,42	4,862 6,662	5,5282 6,9282	6,2102
Модифицированный Эйлера	<i>y_i</i> <i>y'_i</i> <i>y_{i+0,5}</i> <i>y'_{i+0,5}</i>	3 6 3,3 6,1	3,61 6,21 3,9205 6,3205	4,24205 6,44205 4,564152 6,564152	4,898465 6,698465 5,233388 6,833388	5,581803 6,981803 5,930893 7,130893	6,294892
Эйлера—Коши	<i>y_i</i> <i>y'_i</i> <i>y_{i+1}</i> <i>y'_{i+1}</i>	3 6 3,6 6,2	3,61 6,21 4,231 6,431	4,24205 6,44205 4,886255 6,686255	4,898465 6,698465 5,568311 6,968311	5,581803 6,981803 6,279983 7,279983	6,294892
Рунге—Кутта порядка 4	<i>y_i</i> <i>k₁</i> <i>k₂</i> <i>k₃</i> <i>k₄</i>	3 0,6 0,61 0,6105 0,6211	3,610342 0,621034 0,632086 0,632638 0,644298	4,242805 0,644280 0,656494 0,657105 0,669991	4,899716 0,669972 0,683470 0,684145 0,698386	5,583648 0,698365 0,713283 0,714029 0,729768	6,297440
Уточненный Эйлера	<i>y_i</i> <i>y'_i</i>	3 6	3,6 6,2	4,24 6,44	4,888 6,688	5,5776 6,9776	6,28352
Адамса	<i>y_i</i> <i>y'_i</i> <i>q_i</i> <i>-Δq_i</i> <i>Δq_i/2</i> <i>Δ²q_i</i> <i>5Δ²q_i/12</i> <i>Δ³q_i</i> <i>3Δ³q_i/8</i>	3 6 0,6	3,610341 6,210341 0,621034 0,021034 0,010517	4,242805 6,442805 0,644280 0,023246 0,011623 0,002212 0,000922 0,001018 0,000232 0,000087	4,899717 6,699717 0,669972 0,025691 0,012846 0,002445 0,000922 0,001018 0,000232 0,000087	5,583640 6,983640 0,698364 0,028392 0,014196 0,002701 0,001125 0,000256 0,000096	6,297421
Милна	<i>y_i</i> <i>y'_i</i> <i>y_{i+1}</i> <i>y'_{i+1}</i>	3 6	3,610341 6,210341	4,242805 6,442805	4,899717 6,699717 5,583641 6,983641	5,583648 6,983648 6,297433 7,297433	6,297441

При использовании усовершенствованного метода Эйлера — Коши в соответствии с формулой (69в) определяют вспомогательную величину $\bar{y}_{i+1} = y_i + hy'_i$, вычисляют $\bar{y}_{i+1} = f(x_{i+1}, \bar{y}_{i+1})$ и находят очередное значение функции по формуле $y_{i+1} = y_i + (y'_i + \bar{y}'_{i+1})h/2$. Это значение можно уточнять с помощью итерационной процедуры. Для этого выбирают грубое приближение $y_{i+1}^{(0)} = y_i + hf(x_i, y_i)$ и строят итерационный процесс по формуле $y_{i+1}^{(k)} = y_i + (f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k-1)}))h/2$; $k = 0, 1, \dots$, повторяя вычисления до требуемого числа значащих цифр. Если после нескольких итераций требуемая точность не достигается, то уменьшают шаг.

Высокой точностью отличается метод Рунге — Кутта четвертого порядка. При использовании этого метода приращения функции на очередном шаге приравниваются сумме первых четырех членов ряда Тейлора (32), но непосредственное вычисление производных заменяется вычислением составляющих приращения функции в формуле (69г). При вычислениях на микрокалькуляторе эту формулу записывают как $y_{i+1} = (((k_2 + k_3)2 + k_1 + k_4) : 6 + y_i)$ и реализуют вводом одного предложения. Для оценки выбора шага в рассматриваемом методе вычисляют величину $|k_2 - k_3| / |k_1 - k_2|$. Если она превышает несколько сотых, то шаг уменьшают.

Недостаток метода Рунге — Кутта связан с большим числом операций. Точность вычислений при использовании одноступенчатых методов более низкого порядка увеличивается при уменьшении шага до определенного предела, но в этом случае число операций также возрастает. Поэтому при необходимости точного решения дифференциального уравнения часто используют многоступенчатые методы.

Относительно небольшой объем вычислений характерен для двуступенчатого уточненного метода Эйлера, описываемого формулой (70). При использовании этого метода начальное значение приходится определять одним из одноступенчатых методов (обычно методом Эйлера). При использовании многоступенчатых методов высших порядков с помощью одноступенчатых методов определяют несколько начальных значений функции.

При использовании метода Адамса четыре начальные значения функции вычисляют одноступенчатым методом при достаточно малом шаге или по формулам Крылова (71б). Так как точность вычисления начального участка в значительной мере определяет точность последующих результатов, то для определения начальных значений функции в многоступенчатых методах часто используют метод Рунге — Кутта четвертого порядка. Шаг вычислений в методе Адамса выбирают таким, чтобы разности $\Delta^3 q_i$ и $\Delta^3 q_{i+1}$ отличались не более чем на одну-две единицы разряда, определяющую заданную точность. Если используемый микрокалькулятор не имеет регистров памяти для хранения результатов вычислений по формуле (71 а), то их записывают в вычислительный бланк.

При использовании метода прогноза и коррекции Милна для удобства вычислений на микрокалькуляторе формулу прогноза (72а) записывают как

$$\bar{y}_{i+1} = ((y'_i + y'_{i-2}) 2 - y'_{i-1}) 4h : 3 + y_{i-3},$$

а формулу коррекции (72б) как

$$y_{i+1} = (4y'_i + \bar{y}'_{i+1} + y'_{i-1}) h : 3 + y_{i-1},$$

что позволяет реализовать каждую из них вводом одного предложения.

Пример 23. Решить численными методами дифференциальное уравнение, заданное в предыдущем примере.

Для сравнения результатов решения задачи различными методами выберем одинаковый шаг интегрирования $h = 0,1$. Решая дифференциальное уравнение одноступенчатыми методами в соответствии с формулами (69), записываем результаты в вычислительные бланки, приведенные в табл. 18.

При решении заданного уравнения многоступенчатыми методами в качестве начальных выбираем точные значения функции, приведенные в табл. 18. Результаты вычислений записываем в бланки, приведенные в этой же таблице.

Сравнение результатов решения дифференциального уравнения в приведенном примере подтверждает увеличение количества вычислительных операций при повышении точности вычисления искомой функции. Поэтому выбор метода численного решения дифференциальных уравнений в основном определяется требуемой точностью решения.

6. РЕШЕНИЕ СИСТЕМ УРАВНЕНИЙ

Системы линейных уравнений обычно решают численно по схеме единственного деления, требующей минимального числа операций по сравнению с другими численными методами, основанными на исключении переменных. Вычислительный бланк в этом случае составляют в виде таблицы из n частей с $n + 2$ столбцами и $n - k$ строками в каждой k -й части. В первые n столбцов исходной части записывают столбцы матрицы коэффициентов решаемой системы уравнений, а в $(n + 1)$ -й столбец — вектор свободных членов. В дополнительный $(n + 2)$ -й столбец исходной части, называемый контрольным, записывают суммы a_i , $i=1, n+2$ остальных элементов соответствующих строк.

Элементы первой («отмеченной») строки каждой k -й части вычисляют делением элементов первой строки (после отмеченной) предыдущей $(k - 1)$ -й части согласно формуле (81а). Элементы остальных строк k -й части вычисляют по элементам отмеченной строки этой части и элементам неотмеченных строк предыдущей части согласно формуле (81б). В результате прямого хода в отмеченной (и единственной) строке n -й части бланка получают значение $x_n = u_n$, а значения остальных искомых переменных вычисляют в процессе обратного хода по формулам (81г).

Правильность вычислений при прямом и обратном ходах контролируют по значениям элементов контрольного столбца (вычисляемых подобно элементам остальных столбцов), которые должны быть равными суммам остальных элементов соответствующей строки. При этом значения переменных x_i , вычисляемых в процессе обратного хода, должны быть на единицу меньшими элементов контрольного столбца, вычисляемых по формулам обратного хода.

Точность решения системы уравнений определяется одной из норм (87) невязок уравнений, получаемых при подстановке вычисленных значений x_i в исходные уравнения. Если норма невязок относительно велика, то решение можно уточнить по той же схеме. В этом случае вычислительный бланк дополняют ($n + 3$)-м столбцом, в исходную часть которого записывают невязки δ_i уравнений. Повторяя вычисления по формулам прямого и обратного ходов для элементов столбца невязок в остальных частях схемы, получают в результате поправки σ_i переменных, по которым вычисляют уточненные значения $x_i^{(1)} = x_i^{(0)} - \sigma_i$ искомых переменных. При необходимости этот процесс повторяют, уточняя дополнительно полученные результаты.

Из формулы (80а) следует, что при $q_i = 1$ и $q_j = 0$ ($j \neq i$) искомые переменные $x_i = f_{ii}$, где f_{ii} — элементы обратной матрицы. Поэтому вычислительный бланк для обращения матрицы по схеме единственного деления отличается только заменой столбца свободных членов q_i единичной матрицей порядка n , диагональные элементы которой равны единицам, а остальные — нулю. В этом случае в процессе прямого хода вычисляют значения элементов b_{n1}, \dots, b_{nn} обратной матрицы, а по формулам обратного хода (рассматривая каждый столбец единичной матрицы как столбец свободных членов) последовательно вычисляют значения остальных элементов $b_{n-1,1}, \dots, b_{n-1,n}; \dots; b_{11}, \dots, b_{1n}$ обратной матрицы.

При использовании схемы единственного деления для нахождения определителя достаточно вычислить лишь все ведущие элементы для использования формулы (81д). В этом случае столбец свободных членов в бланк можно не записывать.

Пример 24. Решить систему линейных уравнений

$$\begin{bmatrix} 12 & -2 & 0 & -2 \\ 20 & 8 & -6 & 0 \\ 0 & -7 & 13 & -4 \\ -2 & 20 & -4 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

и обратить матрицу ее коэффициентов по схеме единственного деления с помощью 8-разрядного микрокалькулятора с округлением отбрасыванием.

Записав исходные данные в нулевую часть вычислительного бланка (табл. 19), выполняем вычисления по формулам (81) для прямого хода, получая в результате $x_4 = 1,0438201$. По формулам обратного хода последовательно вычисляем $x_3 = -0,0035057$; $x_2 = -0,6029791$; $x_1 = 0,2401398$. Невязки заданных уравнений при подстановке вычисленных корней $\delta_1 = -0,0000044$; $\delta_2 = -0,0000026$; $\delta_3 = 0,0000007$; $\delta_4 = 0,0000020$ при норме вектора невязок $\|\delta\|_m = 0,0000055$.

<i>k</i>	<i>j=1</i>	<i>j=2</i>	<i>j=3</i>	<i>j=4</i>	<i>q</i>	<i>S</i>	$\sigma \cdot 10^6$
0	12 20 0 -2	-2 8 -7 20	0 -6 13 -4	-2 0 -4 12	2 0 0 0	10 22 2 26	-4,4 -2,6 -0,7 2
1	1 0 0 0	-0,1666666 11,333332 -7 19,666667	0 -6 13 -4	-0,1666666 3,333332 -4 11,666667	0,1666666 -3,333332 0 0,333332	0,8333333 5,333334 2 27,666667	-0,3666666 4,7333332 -0,7 1,2666668
2		1 0 0	-0,5294118 9,2941170 6,4117650	0,2941175 -1,9411775 5,8823560	-0,2941175 -2,0588220 6,1176411	0,4705883 5,2941181 18,411764	0,417647 2,223529 -6,947058
3			1 0	-0,2088608 7,2215223	-0,2215188 7,5379705	0,5696203 14,759493	0,2392404 -8,4810108
4				1	1,0438201	2,0438200	-1,1744076
					-0,0035057 -0,6029791 0,2401398	0,9964941 0,3970209 1,2401398	-0,0060473 0,7598593 -0,4357580

Для повышения точности вычислений при уточнении корней умножим значения невязок на 10^6 и запишем результат в нулевую часть бланка. Повторив вычисления по формулам прямого и обратного ходов для столбца поправок, находим $\sigma_1 = -1,1744076 \cdot 10^{-6}$; $\sigma_2 = -6,0473 \cdot 10^{-8}$; $\sigma_3 = 7,598593 \cdot 10^{-7}$; $\sigma_4 = 4,35758 \cdot 10^{-7}$. Следовательно, уточненные значения $x_i^{(1)} = x_i^{(0)} - \sigma_i$ исключенных переменных $x_1 = 0,2401402$; $x_2 = -0,6029799$; $x_3 = -0,0035057$; $x_4 = 1,0438213$. Невязки уравнений для уточненных значений переменных $\delta_1 = 0,0000004$; $\delta_2 = -0,000001$; $\delta_3 = 0$; $\delta_4 = 0$ при норме вектора невязок $\|\delta\|_m = 0,0000011$.

Для обращения матрицы составляем вычислительный бланк (табл. 20), в котором столбец свободных членов заменен единичной матрицей, и по формулам (81) вычисляем элементы обратной матрицы

$$A^{-1} = \begin{bmatrix} 0,1200699 & -0,0202577 & -0,0035056 & 0,0188431 \\ -0,3014896 & 0,1783522 & 0,0744959 & -0,0254162 \\ -0,0017528 & 0,0039439 & 0,0876424 & 0,0289219 \\ 0,5219102 & -0,2992987 & -0,0955301 & 0,1384749 \end{bmatrix}.$$

Для численного решения системы линейных уравнений по компактной схеме вычислительный бланк составляют с двумя частями. В первую часть записывают матрицу коэффициентов и столбец свободных членов исходной системы уравнений. В первый столбец второй части записывают элементы $l_{11} = a_{11}$ первого столбца исходной части, а в первую строку первой части записывают вычисленные делением на a_{11} элементы u_{1j} . Последующие вычисления производятся «елочкой»: по формуле (82а) вычисляют элементы l_{ij} очередного столбца нижней треугольной матрицы и по формулам (82б) — элементы u_{ii} очередной строки верхней треугольной матрицы. По

<i>k</i>	<i>i=1</i>	<i>i=2</i>	<i>i=3</i>	<i>i=4</i>	<i>q₁</i>	<i>q₂</i>	<i>q₃</i>	<i>q₄</i>
0	12	-2	0	-2	1	0	0	0
	20	8	-6	0	0	1	0	0
	0	-7	13	-4	0	0	1	0
	-2	20	-4	12	0	0	0	1
1	1	-0,1666666	0	-0,1666666	0,08335333	0	0	0
	0	11,333332	-6	3,333332	-1,666666	1	0	0
	0	-7	13	-4	0	0	1	0
	0	19,666667	-4	11,666667	0,1666666	0	0	1
2	1	-0,5294118	0,2941175	-0,1470588	0,0882353	0	0	0
	0	9,2941170	-1,9411775	-1,0294116	0,6176471	1	0	0
	0	6,4117650	5,8823560	3,0588230	-1,7352942	0	0	1
3		1	-0,2088608	-0,1107594	0,0664557	0,1075949	0	0
		0	7,2215223	3,7689862	-2,1613925	-0,6898732	0	1
4			1	0,5219102	-0,2992987	-0,0955301	0,1384749	
				-0,0017528	0,0039439	0,0876424	0,0289219	
				-0,3014896	0,1783522	0,0744959	-0,0254162	
				0,1200699	-0,0201577	-0,0035689	0,0188432	

окончании прямого хода получают элементы матрицы LU и свободные члены эквивалентной системы уравнений (82в). При обратном ходе по формулам (82г) вычисляют вспомогательные величины y_i , а по формулам (82д) — искомые переменные x_i .

Компактную схему целесообразно использовать лишь при вычислениях на микрокалькуляторе с дополнительными регистрами памяти. В противном случае необходимость дополнительной записи результатов суммирования в формулах (82) лишает компактную схему ее преимуществ.

Вместе с тем при использовании микрокалькуляторов с дополнительным регистром памяти иногда удобно использовать метод Жордана — Гаусса, позволяющий вычислить искомые переменные согласно формуле (84) в результате только прямого хода. Вычислительный бланк в этом случае подобен бланку для схемы единственного деления, но все части бланка содержат n строк, а номер отмеченной строки для каждой k -й части равен k . Число операций в этом случае не меньше, чем в схеме единственного деления, но они более однотипны, что упрощает вычисления с помощью микрокалькулятора.

Пример 25. Решить систему линейных уравнений, заданную в примере 24, методом LU -разложения (по компактной схеме) и методом исключений Жордана—Гаусса.

Записав исходные данные в первую часть вычислительного бланка (табл. 21), вычисляем элементы второй части по формулам (82) компактной схемы решения.

Составив вычислительный бланк для решения уравнения методом Жордана — Гаусса (табл. 22), записываем исходные данные в ее начальную часть, и, продолжая вычисления по формулам (84), получаем в результате прямого хода значения искомых переменных.

Таблица 21

k	$j=1$	$j=2$	$j=3$	$j=4$	q_i	Σ
1	12	-2	0	-2	2	10
	20	8	-6	0	0	22
	0	-7	13	-4	0	2
	-2	20	-4	12	0	26
2	12	-0,16666666	0	-0,16666666	0,16666666	11,833333
	20	11,333332	-0,5294118	0,2941175	-0,2941175	-10,8419214
	0	-7	9,2941170	-0,2088608	-0,2215188	-0,2215194
	-2	19,666666	6,4117650	7,2215223	1,0438201	2,0438240
					-0,0035057	0,9964941
					-0,6029792	0,3970209
					0,2401398	1,2401398

Таблица 22

k	$j=1$	$j=2$	$j=3$	$j=4$	q	Σ
0	12	-2	0	-2	2	10
	20	8	-6	0	0	22
	0	-7	13	-4	0	2
	-2	20	-4	12	0	26
1	1	-0,16666666	0	-0,16666666	0,16666666	0,83333333
	0	11,333332	-6	3,333332	-3,333332	5,333334
	0	-7	13	-4	0	2
	0	19,666667	-4	11,666667	0,333332	27,666667
2		0	-0,0882352	-0,1176470	0,1176470	0,91176461
		1	-0,5294118	0,2941175	-0,2941175	0,4705881
		0	9,2941170	-1,9411775	-2,0588220	5,2941181
		0	6,4117650	5,8823560	6,1176441	15,411764
3			0	-0,1360758	0,0981012	0,9620251
			0	0,1835441	-0,4113921	0,7721521
			1	-0,2088608	-0,2215188	0,5696203
			0	7,2215223	7,5379705	14,759493
4				0	0,2401398	1,2401398
				0	-0,6029791	0,3970209
				0	-0,0035057	0,9964941
				1	1,0438201	2,0438201

При решении плохо обусловленной системы линейных уравнений (со значительным различием в величинах коэффициентов, близостью нулю определителя или существенной асимметрией матрицы коэффициентов) операционная погрешность точных методов решения может оказаться недопустимо большой. В этих случаях целесо-

сообразно использовать приближенные численные или буквенные методы решения.

Преимущества итерационных численных методов решения систем линейных уравнений связаны с независимостью их окончательного результата от погрешностей вычисления промежуточных результатов. Итерационный процесс вычисления искомых переменных оканчивают, когда в очередных их значениях совпадает требуемое число значащих цифр.

Системы уравнений, не удовлетворяющие условиям сходимости методов простых итераций или Зейделя, можно эквивалентно преобразовать в системы, удовлетворяющие этим требованиям. Однако в связи с отсутствием четкого алгоритма такие преобразования для общего случая могут оказаться громоздкими и потребовать больших затрат времени. Поэтому при использовании микрокалькуляторов иногда более эффективными оказываются релаксационные методы численного решения.

При использовании метода простых (координатных) релаксаций в вычислительный бланк записывают невязки уравнений при нулевом приближении (обычно принимают $x_i^{(0)} = 0$) и последовательно сводят к нулю наибольшие из этих невязок, записывая в бланк необходимое для этого значение поправки σ_i^k . Подобный процесс повторяют до тех пор, пока величины невязок не достигнут требуемой степени малости. После этого искомые значения x_i вычисляют как суммы соответствующих поправок σ_i^k .

Пример 26. Решить систему уравнений, заданную в примере 24, приближенными численными методами с точностью до трех верных цифр после запятой.

Заданная система уравнений не удовлетворяет условию (87) сходимости метода последовательных приближений (простых итераций). Переславив уравнения системы и используя их линейную комбинацию, получаем эквивалентную систему уравнений

$$\begin{bmatrix} 20 & 8 & -6 & 0 \\ -2 & 20 & -4 & 12 \\ 0 & -7 & 13 & -4 \\ 4 & 18 & 10 & 44 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 36 \end{bmatrix},$$

удовлетворяющую условию сходимости.

Составляем вычислительный бланк (табл. 23), в нулевую часть которого записываем элементы системы уравнений, приведенной к нормализованному виду (85). В начальную строку первой части бланка записываем нулевые приближения $x_i^0 = \beta_i$. Элементы остальных строк вычислительного бланка (первая часть табл. 23) вычисляем по формуле

$$x_i^{(k)} = \sum_{j=1}^4 \alpha_{ij} x_j^{k-1},$$

где в качестве элементов α_{ij} используем коэффициенты i -й строки нулевой части бланка, а значения x_j берем из предыдущей ($k-1$)-й строки первой части. Для вычислений по приведенной формуле особенно удобны микрокалькуляторы.

p	k	x_1	x_2	x_3	x_4	β_t
0	x_1	0	-0,4	0,3	0	0
	x_2	0,1	0	0,2	-0,6	0
	x_3	0	0,5384615	0	0,3076923	0
	x_4	-0,090909	-0,4090909	-0,2272727	0	0,8181818
1	0	0	0	0	0,8181818	
	1	0	-0,490909	0,2517482	0,8181818	
	2	0,271888	-0,4405594	-0,0125873	0,9617928	
	3	0,1724476	-0,5524042	0,058712	0,9765543	
	4	0,2385752	-0,5569454	0,0030299	1,0151447	
	5	0,223687	-0,5846234	0,0124586	1,0236458	
	6	0,2375868	-0,589327	0,0001708	1,0341794	
	7	0,235782	-0,5967149	0,0008791	1,0374717	
	8	0,2389496	-0,598729	-0,002086	1,0406568	
	9	0,2388658	-0,6009171	-0,0021901	1,0418677	
	10	0,2397098	-0,6016721	-0,0029961	1,0427942	
	11	0,23977	-0,6023048	-0,0031175	1,0432095	
	12	0,2399867	-0,6025722	-0,0033304	1,0435192	
	13	0,2400297	-0,6027789	-0,0033791	1,0436286	
	14	0,2400978	-0,60285	-0,0034568	1,0437202	
2	0	0	0	0	0,8181818	
	1	0	-0,490909	-0,0125873	1,0218689	
	2	0,1925875	-0,59638	-0,0067065	1,0461717	
	3	0,2365401	-0,6053903	-0,0040804	1,0452651	
	4	0,240932	-0,6038818	-0,003547	1,0441276	
	5	0,2404886	-0,6031371	-0,0034961	1,0438516	
	6	0,240206	-0,6029895	-0,0035016	1,0438183	

куляторы, имеющие дополнительный регистр памяти. Искомые значения переменных с заданной точностью получаем на 14-й итерации.

При использовании метода Зейделя вычисления выполняются по формуле

$$x_i^{(k)} = \sum_{j=1}^{i-1} \alpha_{ij} x_j^k + \sum_{j=1}^4 \alpha_{ij} x_j^{(k-1)}.$$

Записывая результаты вычислений по этой формуле во вторую часть табл. 23, получаем требуемую точность всего за шесть итераций.

Для применения метода координатных релаксаций приводим исходную систему уравнений к нормализованному виду

$$\begin{aligned} -x_1 + x_2/6 + x_4/6 + 1/6 &= 0; \\ -2,5x_1 - x_2 + 0,75x_3 &= 0; \\ 7x_2/13 - x_3 + 4x_4/13 &= 0; \\ x_1/6 - 5x_2/3 + x_3/3 - x_4 &= 0. \end{aligned}$$

В верхнюю часть вычислительного бланка (табл. 24) записываем транспонированную матрицу γ коэффициентов при неизвестных решаемой системы уравнений, а в нулевую строку основной части бланка — невязки каждого из

T	$\frac{1}{1/6}$ 0 $1/6$	$-2,5$ -1 $0,75$ 0	0 $7/13$ -1 $4/13$	$1/6$ $-5/3$ $1/3$ -1
i	δ_1	δ_2	δ_3	δ_4
0	<u>0,1666666</u>	0	0	0
1	0	<u>-0,4166665</u>	0	0,0277777
2	-0,0694444	0	-0,2243588	<u>0,7222218</u>
3	<u>0,0509259</u>	0	-0,0021368	0
4	0	<u>-0,1273147</u>	-0,0021368	0,0084876
5	-0,0212191	0	-0,0706908	<u>0,2206787</u>
6	<u>0,0155606</u>	0	-0,0027897	0
7	0	<u>-0,0389015</u>	-0,0027897	0,0025934
8	-0,0064835	0	-0,0237366	<u>0,0674292</u>
9	<u>0,0047547</u>	0	-0,0029892	0
10	0	<u>-0,0118867</u>	-0,0029892	0,0007924
11	-0,0019811	0	-0,0093897	<u>0,0206035</u>
12	0,0014528	0	<u>-0,0030502</u>	0
13	0,0014528	<u>-0,0022876</u>	0	-0,0010167
14	0,0010716	0	-0,0012317	<u>0,0027959</u>
15	<u>0,0015375</u>	0	-0,0003715	0
16	0	<u>-0,0038437</u>	-0,0003715	0,0002562
17	-0,0006406	0	-0,0024411	<u>0,0066623</u>
18	<u>0,0004697</u>	0	-0,0003912	0
19	0	<u>-0,0011742</u>	-0,0003912	0,0000782
20	-0,0001957	0	-0,0010234	<u>0,0020352</u>
21	0,0001435	0	<u>-0,0003972</u>	0
22	0,0001435	<u>-0,0002979</u>	0	-0,0001324
23	0,0000939	0	-0,0001604	<u>0,0003641</u>
24	<u>0,0001545</u>	0	-0,0000484	0
25	0	<u>-0,0003862</u>	-0,0000484	0,0000257
26	-0,0000643	0	-0,0002563	<u>0,0006693</u>
27	0,0000472	0	<u>-0,0000504</u>	0
28	<u>0,0000472</u>	-0,0000378	0	-0,0000168
x_j	0,2401167	-0,602759	-0,0034978	1,04346

уравнений при $x_i^{(0)} = 0$, равные свободным членам нормализованной системы уравнений.

Принимая в качестве поправки $\delta x_i = \max |\delta_i|$ наибольшую по модулю невязку с обратным знаком, последовательно вычисляем невязки остальных уравнений, умножая выбранную величину δx_i на коэффициент a_{ij} каждого из остальных столбцов и складывая полученные значения с невязками предыдущей строки. Подчеркиваем невязки, устраниемые на каждой итерации. Суммы подчеркнутых значений невязок в каждом столбце равны искомым переменным с требуемой точностью.

Решение систем линейных уравнений по расчетным формулам, полученным буквенными методами, особенно удобно в случае часто встречающихся в инженерных задачах систем линейных уравнений с разреженными матрицами A и векторами Q свободных членов. В точных методах численного решения систем линейных уравнений, подобных схеме единственного деления, результат каждого деления элементов преобразуемой матрицы на ведущий, главный или опорный элемент может оказаться бесконечной десятичной дробью, что приводит к погрешности округления. С увеличением числа операций погрешность округления накапливается и снижает точность результатов. При использовании буквенных методов расчетные формулы оказываются точными, а число операций деления при вычислениях по этим формулам мало, что существенно снижает операционную погрешность.

Вместе с тем при использовании буквенных методов может возникнуть погрешность, связанная с вычитанием близких или одинаковых по величине членов определителя с противоположными знаками. При вычислениях на ЭВМ в автоматическом режиме эта погрешность может привести к значительной потере точности результата. При вычислениях на непрограммируемых микрокалькуляторах полученные результаты визуально контролируются по индикатору и вычислительному бланку, что дает возможность избежать возникновения операционной погрешности подобного вида. Кроме того, в процессе составления формул буквенными методами при решении инженерных задач можно использовать специальные алгоритмы [25] для устранения одинаковых по величине, но противоположных по знаку членов определителя или их составляющих.

Пример 27. Решить систему уравнений

$$\begin{bmatrix} 3,001 & 0 & 0 & -0,001 \\ 20 & 1,6 & 0 & 0 \\ 0 & 20 & 1,5 & 0 \\ -0,001 & 0 & -20 & 1,111 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Решая заданную систему уравнений по схеме единственного деления при помощи 8-разрядного микрокалькулятора с фиксированной запятой и с округлением отбрасыванием, получаем $x_1 = 1109,4097$; $x_2 = -13859,292$; $x_3 = 184901,51$; $x_4 = 3328561,2$ с невязками $\delta_1 = -0,2227$; $\delta_2 = 13,327$; $\delta_3 = 166,42$; $\delta_4 = 0,1$ при норме $\|\delta\|_m = 166,9529$.

В связи с разреженностью матрицы коэффициентов и вектора свободных членов заданной системы уравнений целесообразно использовать для более

точного решения буквенные методы. В соответствии с формулой (96) вычисляем

$$\Delta = \det \begin{bmatrix} 1 & 1 & 4 \\ 2 & 2 & 1 \\ 3 & 3 & 2 \\ 4 & 4 & 3 \end{bmatrix}_{\text{mod } 2} = \det \begin{bmatrix} 1 & 1 & 4 \\ 2 & 2 & 1 \\ 3 & 3 & 2 \\ 4 & 4 & 1 \end{bmatrix} = a_{11}a_{22}a_{33}a_{44} - a_{14}(a_{22}a_{33}a_{41} + a_{21}a_{32}a_{43}) = 0,001864;$$

$$\Delta_{11} = \det \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 4 & 4 \end{bmatrix}_{\text{mod } 2} = \det \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 4 & 4 \end{bmatrix} = a_{22}a_{33}a_{44} = 2,6664;$$

$$\Delta_{12} = -\det \begin{bmatrix} 1 & 1 \\ 3 & 3 \\ 4 & 1 \\ 1 & 3 \end{bmatrix}_{\text{mod } 2} = -\det \begin{bmatrix} 2 & 1 \\ 3 & 3 \\ 4 & 4 \end{bmatrix} = -a_{21}a_{33}a_{44} = -33,33;$$

$$\Delta_{13} = \det \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 4 & 1 \\ 1 & 4 \end{bmatrix}_{\text{mod } 2} = \det \begin{bmatrix} 1 & 2 \\ 2 & 2 \\ 4 & 4 \end{bmatrix} = a_{21}a_{32}a_{44} = 444,4;$$

$$\Delta_{14} = -\det \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 4 & 1 \\ 1 & 3 \end{bmatrix}_{\text{mod } 2} = -\det \begin{bmatrix} 2 & 1 & 2 \\ 3 & 2 & 3 \\ 4 & 3 & 1 \end{bmatrix} = -(a_{21}a_{33}a_{43} + a_{22}a_{31}a_{41}) = 8000,0024.$$

В соответствии с формулой (80а) находим

$$x_1 = \Delta_{11}/\Delta = 1430,4721; \quad x_2 = \Delta_{12}/\Delta = -17880,901; \quad x_3 = \Delta_{13}/\Delta = 238412,01; \\ x_4 = \Delta_{14}/\Delta = 4291846,7$$

при невязках уравнений после подстановки вычисленных значений искомых переменных $\delta_1 = 0$; $\delta_2 = 0,001$; $\delta_3 = 0,01$; $\delta_4 = 0$. Таким образом, норма невязок $\|\delta\|_m = 0,01004987$ при использовании буквенного метода на четыре порядка меньше нормы невязок при использовании точного метода исключений переменных.

Буквенные методы особенно удобны при решении системы уравнений с изменяющимися в заданных интервалах элементами матрицы коэффициентов, вектора свободных членов, их составляющими или множителями. В частности, в инженерной практике при отображении системы линейных дифференциальных уравнений в область комплексной частоты $p = \sigma + j\omega$ возникает задача вычисления коэффициентов при различных степенях p дробно-рациональных функций (66). При решении подобных задач составляют буквенные разложения определителей, равных числителю и знаменателю анализируемой функции, после чего подставляют известные численные значения элементов в полученные формулы.

При разложениях определителей по буквенным символам составляющих элементов модели для приведения подобных членов удобно использовать алгоритмы метода обобщенных чисел [23]. В частности, при наличии идентичных составляющих в нескольких элементах матрицы коэффициентов системы уравнений после составления по формуле (96) контурного числа каждый его элемент m_l можно заменить множеством составляющих соответствующего элемента a_{ml} матрицы, после чего найти произведения этих множеств над полем модуля 2*, т. е. образовать сочетания (столбцы) из n

* В алгебре модуля 2 результат любой операции над двумя идентичными объектами равен нулю (пустому множеству).

различных составляющих. Столбцы с одинаковыми сочетаниями составляющих необходимо вычеркнуть.

При переходе от полученного подобным образом частично-упорядоченного множества (древесного числа) составляющих к обычной форме записи определителя линии, отделяющие умножаемые множества, заменяют скобками, а каждое множество составляющих — их суммой. При необходимости можно предварительно сгруппировать столбцы по определенным признакам, например, по одинаковому числу множителей определенного типа.

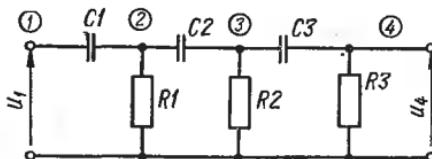


Рис. 13. Схема электрической цепи

Пример 28. Вычислить коэффициенты при степенях комплексной частоты p коэффициента передачи напряжения $K_{41} = u_4/u_1$ электрической цепи, схема которой показана на рис. 13, при $C_1 = C_2 = C_3 = 100 \cdot 10^{-12} \Phi$; $g_1 = g_2 = 1/10 \text{ кОм} = 100 \text{ мКСм}$; $g_3 = -1/2 \text{ кОм} = 500 \text{ мКСм}$.

Составим систему уравнений узловых напряжений с указанной на схеме нумерацией узлов

$$\begin{bmatrix} pC_1 & -pC_1 & 0 & 0 \\ -pC_1 & pC_1 + pC_2 + g_1 & -pC_2 & 0 \\ 0 & -pC_2 & pC_2 + pC_3 + g_2 & -pC_3 \\ 0 & 0 & -pC_3 & pC_3 + g_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix}.$$

Для определения искомого коэффициента передачи напряжения $K_{41} = u_4/u_1 = \Delta_{14}/\Delta_{11}$ находим в соответствии с формулой (96) и описанным выше способом приведения подобных

$$\Delta_{14} = -\det 3 \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & \\ 4 & 3 & \end{bmatrix}_{\text{mod } 2} = -\det 3 \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} = -\begin{bmatrix} -pC_1 \\ -pC_2 \\ -pC_3 \end{bmatrix} = p^3 C_1 C_2 C_3;$$

$$\Delta_{11} = \det 3 \begin{bmatrix} 2 & 3 \\ 2 & 3 & 4 \\ 4 & 3 & 4 \end{bmatrix}_{\text{mod } 2} = \det 3 \begin{bmatrix} 2 & 3 \\ 3 & 4 & 2 \\ 4 & 3 & 4 \end{bmatrix}.$$

Для симметричных матриц можно ограничиться определением произведения над полем модуля 2 только множеств составляющих диагональных элементов матрицы при буквенном разложении ее определителя или главного (симметричного) минора. Поэтому

$$\begin{aligned} \Delta_{11} = 3 \begin{bmatrix} pC_1 pC_2 g_1 \\ pC_2 pC_3 g_2 \\ pC_3 g_3 \end{bmatrix} &= \begin{bmatrix} pC_1 pC_1 & pC_2 & pC_1 pC_2 g_1 & pC_1 pC_2 & g_1 & g_1 & g_1 \\ pC_2 pC_2 pC_3 pC_3 & g_2 & g_2 & pC_2 & g_2 & pC_2 pC_3 & g_2 \\ pC_3 pC_3 pC_3 & g_3 & g_3 & pC_3 & g_3 & g_3 & pC_3 \end{bmatrix} = \\ &= p^3 C_1 C_2 C_3 + p^2 [g_3 (C_1 (C_2 + C_3) + C_2 C_3) + C_1 C_3 g_2 + \\ &+ g_2 C_2 C_3 + g_1 C_2 C_3] + p [(C_1 + C_2) g_2 g_3 + (C_2 + C_3) g_1 g_3 + \\ &+ g_1 g_2 C_3] + g_1 g_2 g_3. \end{aligned}$$

Подставляя численные значения параметров схемы, получаем искомое выражение коэффициента передачи напряжения с численными значениями коэффициентов при степенях p

$$\begin{aligned} K_{41} &= \frac{10^{-30} p^3}{10^{-80} p^3 + 18 \cdot 10^{-24} p^2 + 21 \cdot 10^{-18} p + 5 \cdot 10^{-12}} = \\ &= \frac{p^8}{p^3 + 18 \cdot 10^6 p^2 + 21 \cdot 10^{12} p + 5 \cdot 10^{18}}. \end{aligned}$$

Вычисление переменных x_i или элементов f_{ji} обратной матрицы коэффициентов системы уравнений при изменении элементов a_{ij} матрицы или составляющих q_i вектора свободных членов можно упростить, предварительно составив расчетные формулы. В таких формулах изменяемые величины отображаются буквенными символами, а влияние элементов системы уравнений с фиксированными значениями отображается численными коэффициентами. Если изменяется несколько параметров, то числители и знаменатели функций f_{ji} в формулах (80) раскладывают по формуле (97а), которую удобно записывать в виде

$$\Delta = a_1(\Delta_1^0 + \sum a_r \Delta_{1r}^0 + \dots) + \Delta^0 + \sum a_r \Delta_r^0 + \dots, \quad (124)$$

где a_1 — один из изменяющихся параметров.

При вычислениях по этой формуле с помощью микрокалькулятора задают фиксированные значения параметров a_2, a_3, \dots, a_m в интервалах их изменения и для ряда дискретных значений a_1 определяют соответствующие значения определителей, входящих в выражения для x_i или f_{ji} . Повторяя вычисления для нового набора фиксированных значений параметров a_2, \dots, a_m , получают в результате табличную или графическую модель семейства характеристик $x_i = x_i(a_1)$ или $f_{ji} = f_{ji}(a_1)$ при $a_2 = \text{const}, \dots, a_m = \text{const}$.

Пример 29. Составить расчетную формулу для определения зависимости коэффициента передачи напряжения $K = u_5/u_1$ слабосигнального усилителя, схема которого приведена на рис. 14, от значений g_1 и g_2 резисторов R_1 и R_2 . Влияние конденсаторов пренебрежимо мало, сопротивления резисторов указаны на схеме, параметры обоих транзисторов одинаковы и заданы неопределенной матрицей [24] проводимостей (в мкСм)

$$\begin{bmatrix} g_{66} & g_{69} & g_{6k} \\ g_{96} & g_{99} & g_{9k} \\ g_{k6} & g_{k9} & g_{kk} \end{bmatrix} = \begin{bmatrix} 1000 & -998 & -2 \\ -41000 & 41098 & -98 \\ 40000 & -40100 & 100 \end{bmatrix}.$$

Составим каноническую систему уравнений узловых напряжений для заданной схемы в соответствии с указанной на схеме нумерацией независимых узлов (входов)

$$\begin{bmatrix} g_3 + g_4 + g_{66} & g_{6k} & g_{69} & 0 & 0 \\ g_{k6} & g_{kk} + g_{66} + g_5 & g_{k9} & g_{69} & g_{6k} \\ g_{96} & g_{9k} & g_{99} + g_1 + g_2 + g_8 & -g_2 & 0 \\ 0 & g_{96} & -g_2 & g_{99} + g_7 + g_2 & g_{9k} \\ 0 & g_{k6} & 0 & g_{k9} & g_{kk} + g_6 + g_9 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix}$$

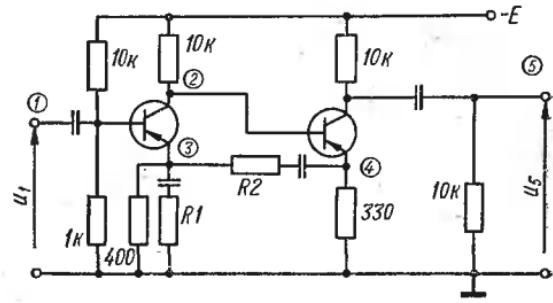


Рис. 14. Схема транзисторного усилителя

Подставляя известные значения проводимостей транзисторов и $g_i = 1/r_i$ резисторов, получаем матрицу проводимостей (в мкСм)

$$\begin{vmatrix} 2100 & -2 & -998 & 0 & 0 \\ 40000 & 1200 & -40100 & -998 & -2 \\ -41000 & -98 & 43598 + g_1 + g_2 & -g_2 & 0 \\ 0 & -41000 & -g_2 & 44098 + g_2 & -98 \\ 0 & 40000 & 0 & -40100 & 300 \end{vmatrix}.$$

В соответствии с формулами (80), (97) и (124) искомый коэффициент передачи напряжения

$$K = \frac{u_5}{u_1} = \frac{\Delta_{15}}{\Delta_{11}} = \frac{g_1(\Delta_{15, 33}^0 + g_2\Delta_{15, 44}^0) + \Delta_{15}^0 + g_2\Delta_{15, 3+4}^0(3+4)}{g_1(\Delta_{11, 33}^0 + g_2\Delta_{11, 44}^0) + \Delta_{11}^0 + g_2\Delta_{11, 3+4}^0(3+4)}.$$

Вычислив буквенными или численными методами входящие в полученную формулу определители и алгебраические дополнения, получим расчетное выражение

$$K = \frac{g_1(4,793 + 0,0016g_2) + 11960 + 8,9646g_2}{g_1(0,00304 + 0,00000044g_2) + 95,81 - 0,315g_2}.$$

Задаваясь фиксированными значениями g_2 в интервале возможных изменений этой проводимости, можно вычислить по составленной формуле характеристики $K = K(g_2)$ при $g_2 = \text{const}$, записывая вычисляемые значения в бланк или непосредственно на чертеже плоскости K, g_1 для построения семейства характеристик в графической форме.

Решение систем нелинейных уравнений с помощью микрокалькуляторов связано с громоздкими вычислениями, особенно в тех случаях, когда в уравнениях содержатся символы элементарных или специальных функций, операторы которых отсутствуют во входном языке микрокалькулятора используемого типа.

Если заданную систему нелинейных уравнений можно привести к виду (109), то для ее решения может оказаться целесообразным использование метода простых итераций, описываемого формулой (110), или метода Зейделя. В последнем случае вычисленная на очередной итерации искомая переменная x_i^{k+1} подставляется в $(i+1)$ -е уравнение и итерационный процесс образует замкнутый цикл, так как текущее значение $x_n^{(k)}$ на следующей итерации подставляется в первое уравнение. Следовательно, отыскание корней сводится к последовательности вычисления текущих значений переменных

$$\dots x_i^{(k)} = f_i(\dots x_i^{(k-1)}, x_{i+1}^{(k-1)} \dots); x_{i+1}^{(k)} = f_{i+1}(\dots x_i^{(k)}, x_{i+1}^{(k-1)} \dots)$$

до совпадения текущих значений на двух очередных итерациях до требуемого числа значащих цифр.

Пример 30. Вычислить с максимальной точностью корни уравнений

$$x_1 = 0,15e^{0,5x_1} + 0,6e^{0,1x_2};$$

$$x_2 = 0,7e^{0,5x_1} + e^{0,1x_2}$$

на 7-разрядном микрокалькуляторе со стандартной формой представления чисел и округлением отбрасыванием.

Для удобства вычислений опишем решение заданной системы уравнений методом Зейделя формулами:

$$0,6e^{0,1x_2} = F1; \quad 0,15e^{0,5x_1} + F1 = x_1; \quad 0,7e^{0,5x_1} = F2; \quad e^{0,1x_2} + F2 = x_2.$$

В этом случае вычисленное значение x_1 используется для вычисления $F2$, а значение x_2 , хранящееся в индикаторном регистре, используется для вычисления $F1$ на следующей итерации. В вычислительный блок заносятся текущие значения $F1$, x_1 , $F2$ и x_2 . При начальном приближении $x_1^{(0)} = x_2^{(0)} = 0$ максимальная точность вычисления корней системы уравнений заданного вида достигается на 11-й итерации (табл. 25).

Таблица 25

k	$F1$	x_1	$F2$	x_2
0		0		0
1	0,6	0,75	1,018493	2,018493
2	0,7341979	0,9524465	1,126987	2,350650
3	0,7589944	1,0004910	1,154388	2,419378
4	0,7642288	1,011597	1,160816	2,434531
5	0,7653877	1,014134	1,162289	2,437935
6	0,7656482	1,014710	1,162624	2,438704
7	0,7657071	1,014840	1,162700	2,438879
8	0,7657205	1,014870	1,162717	2,438918
9	0,7657236	1,014877	1,162721	2,438927
10	0,7657242	1,014878	1,162722	2,438929
11	0,7657243	1,014879	1,167722	2,438930
12	0,7657243	1,014879	1,167722	2,438930

Необходимо подчеркнуть, что не существует универсального метода решения нелинейных уравнений любого вида. Иногда решение даже наиболее распространенным методом сходится медленно или расходится. Кроме того, результат вычислений может сходиться к корням уравнения, не имеющим физического смысла или не представляющим интереса по условиям задачи. Поэтому при решении многих систем нелинейных уравнений выбор метода, обеспечивающего достаточно быструю сходимость к искомым корням, связан с большими трудностями. Вместе с этим при инженерных расчетах проверка сходимости выбранного метода решения часто облегчается при учете физических условий задачи.

7. ГАРМОНИЧЕСКИЙ И СПЕКТРАЛЬНЫЙ АНАЛИЗ

Аппроксимация функции рядом Фурье (гармонический анализ) имеет практический смысл (если не является необходимой по условиям задачи) в том случае, когда заданная функция представима с требуемой точностью суммой небольшого числа гармоник, а используемый микрокалькулятор автоматически вычисляет тригонометрические функции.

Интерполяция табличной модели с равномерным шагом и $n + 1$ равноотстоящими узлами достигается при использовании усеченного ряда (30) с n коэффициентами, который при четном n имеет вид

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\frac{n}{2}-1} \left(a_k \cos \frac{2\pi kx}{n} + b_k \sin \frac{2\pi kx}{n} \right) + \frac{a_{n/2}}{2} \cos \pi x, \quad (125a)$$

а при нечетном

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\frac{n-1}{2}} \left(a_k \cos \frac{2\pi kx}{n} + b_k \sin \frac{2\pi kx}{n} \right). \quad (125b)$$

Такой ряд приближает аппроксимируемую функцию наилучшим, в смысле среднеквадратичного отклонения, образом между узлами модели в заданном интервале изменения аргумента.

При гармоническом анализе функций, заданных графиком или аналитическим выражением, предварительно разбивают интервал T аргумента на n равных отрезков длиной T/n каждый и определяют значения $y_i = y(x_i)$ в точках разбиения и на границах исходного интервала аргумента. Полученная таким образом табличная модель имеет $n + 1$ равноотстоящих узлов x_i с номерами $i = 0, 1, \dots, n$. С увеличением числа узлов точность представления функции табличной моделью повышается, но возрастает время последующих вычислений.

Коэффициенты рядов (125) вычисляют согласно формулам (31a) по значениям y_i функции в узлах ее табличной модели с номерами $i = 0, 1, \dots, n - 1$. При гармоническом анализе предполагается, что заданная функция периодическая и ее значения $y(x) = y(x + aT)$ повторяются при изменении аргумента на величину aT , кратную периоду T . Следовательно, $y_n = y_0$ и поэтому в формулах (31a) значение y_n отдельно не учитывается.

Если периодическая функция на границе интервала изменяется скачком (рис. 15, a), то возникает погрешность ее интерполяции рядами (125), так как вычисленное значение $y_n = y(x_n + ax) = y_0$ отличается от истинного значения $y_n = y(x_n - ax)$. Эта погрешность, как и другие погрешности, связанные с аппроксимацией кусочно-непрерывных функций, уменьшаются при увеличении числа n разбиений исходного интервала аргумента, но в этом случае возрастает число членов тригонометрического ряда (гармонических составляющих) и затраты времени на вычисления.

В тех случаях, когда ряд Фурье используют для аппроксимации функций (или интерполяции их табличных моделей), значения которых за пределами заданного интервала аргумента не представляют интереса, можно формально рассматривать их как периодические и использовать ряды (125) в пределах заданного интервала. Если

значения y_0 и y_n на границах заданного интервала для таких функций не совпадают, то можно дополнить их табличные модели ($n + 2$ -м) равноотстоящим узлом x_{n+1} (рис. 15, б) со значением функции $y(x_{n+1}) = y_0$. Заменив в формулах (31а) число n числом $n + 1$, получаем ряд Фурье вида (125), сумма членов которого при $x = x_n$ равна значению $y_n = y(x_n)$ исходной функции. Погрешность интерполяции в других точках заданного интервала, отличающихся от узловых, зависит от вида функции.

Для достаточно гладких функций часто ограничиваются четырьмя отрезками разбиения интервала аргумента (пятью узлами таблич-

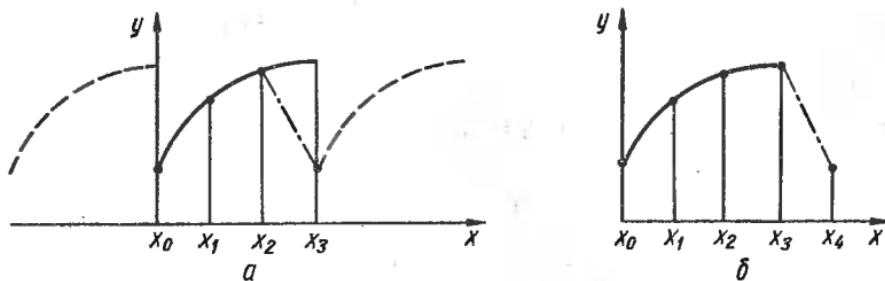


Рис. 15. Координаты узлов табличной модели при гармоническом анализе периодической (а) и непериодической (б) функций

ной модели). В этом случае коэффициенты ряда Фурье (125а) на микрокалькуляторе удобно вычислять по схеме, приведенной на рис. 16, а и отличающейся от описанных в литературе [14, 29]. Схемы вычислений коэффициентов ряда Фурье по 8 и 12 ординатам показаны соответственно на рис. 16, б и в. Подобные схемы нетрудно модифицировать для использования микрокалькуляторов с дополнительными регистрами памяти или табличных моделей с другим числом узлов.

Пример 31. Аппроксимировать усеченным рядом Фурье табличную модель функции с ординатами $y_i = 8,0; 8,3; 12,4; 18,1; 20,2; 16,0$, заданную в примере 17.

По числу узлов табличной модели с дополнительным узлом для $y_n = 8,0$ заданная функция интерполируется рядом с шестью коэффициентами

$$y(x) \approx \frac{a_0}{2} + a_1 \cos \frac{\pi x}{3} + b_1 \sin \frac{\pi x}{3} + a_2 \cos \frac{2\pi x}{3} + b_2 \sin \frac{2\pi x}{3} + a_3 \cos \pi x.$$

Составив в соответствии с формулами (31а) схему, аналогичную показанным на рис. 16, а—в, вычисляем коэффициенты ряда

$$\begin{aligned} a_0 &= (y_0 + y_1 + y_2 + y_3 + y_4 + y_5) : 3 = 27,66666; \\ a_1 &= ((y_0 - y_3) \cdot 2 + y_1 + y_5 - y_2 - y_4) : 6 = -4,75; \\ a_2 &= (y_0 + y_3 - a_0) : 2 = -0,78333; \\ a_3 &= (y_0 + y_2 + y_4 - y_1 - y_3 - y_5) : 6 = -0,3; \\ b_1 &= (2a_3 - a_1 + y_1 - y_4) : \sqrt{3} = -4,474464; \\ b_2 &= (y_1 + y_4 - a_0 + a_2) : \sqrt{3} = 0,028868. \end{aligned}$$

При усечении полученного интерполирующего многочлена отбрасыванием третьей гармоники получим приближение с максимальной относительной погрешностью около 5% при $x_1 = 1$, а при дополнительном отбрасывании второй гармоники — приближение с максимальной относительной погрешностью около 14% при $x_1 = 0$.

Приведенные выше формулы иллюстрируют возможность модификации вычислительных схем, показанных на рис. 16, *a* — *b*, с целью уменьшения числа вводов исходных данных.

Если в частотную область с помощью преобразования (2) отображается периодическая функция $f(t) = f(t + aT)$, где a — целое число и T — период повторения, то при задании функции $f(t)$ табличной моделью с $n + 1$ узлами она отображается в частотную область множеством (дискретным спектром) гармонических составляющих рядов (125). При аналитическом задании функции $f(t)$ точность ее спектрального представления определяется количеством членов аппроксимирующего ряда (125). В тех случаях, когда функция $f(t)$ непериодична и равна нулю за пределами интервала $t_a \leq t \leq t_b$, задача спектрального анализа сводится к приближенному вычислению вещественной и мнимой составляющих (косинус- и синус-преобразований Фурье)

$$A(\omega) = \int_{t_a}^{t_b} f(t) \cos \omega t dt; \quad B(\omega) = \int_{t_a}^{t_b} f(t) \sin \omega t dt \quad (126a)$$

спектральной плотности (2).

На практике часто удобнее смещать начало отсчета в точку t_0 (обычно располагаемую в средине интервала аргумента) и использовать нормировки $x = (t - t_0)/\tau$; $\nu = \omega t$; $x_0 = t_0/\tau$, где τ — удобная масштабная единица, например, $\tau = t_b - t_a$ или $\tau = (t_b - t_a)/2$. В этом случае

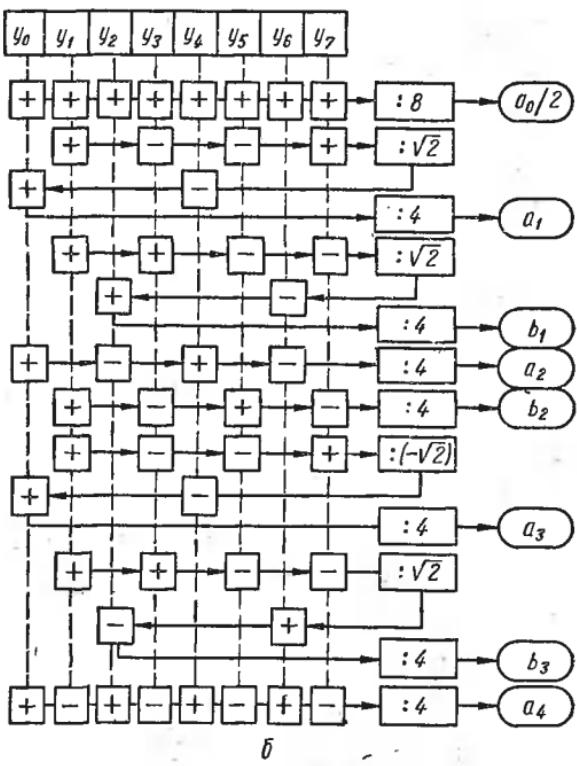
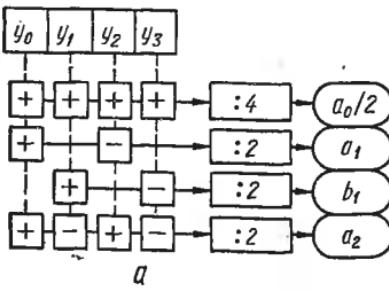


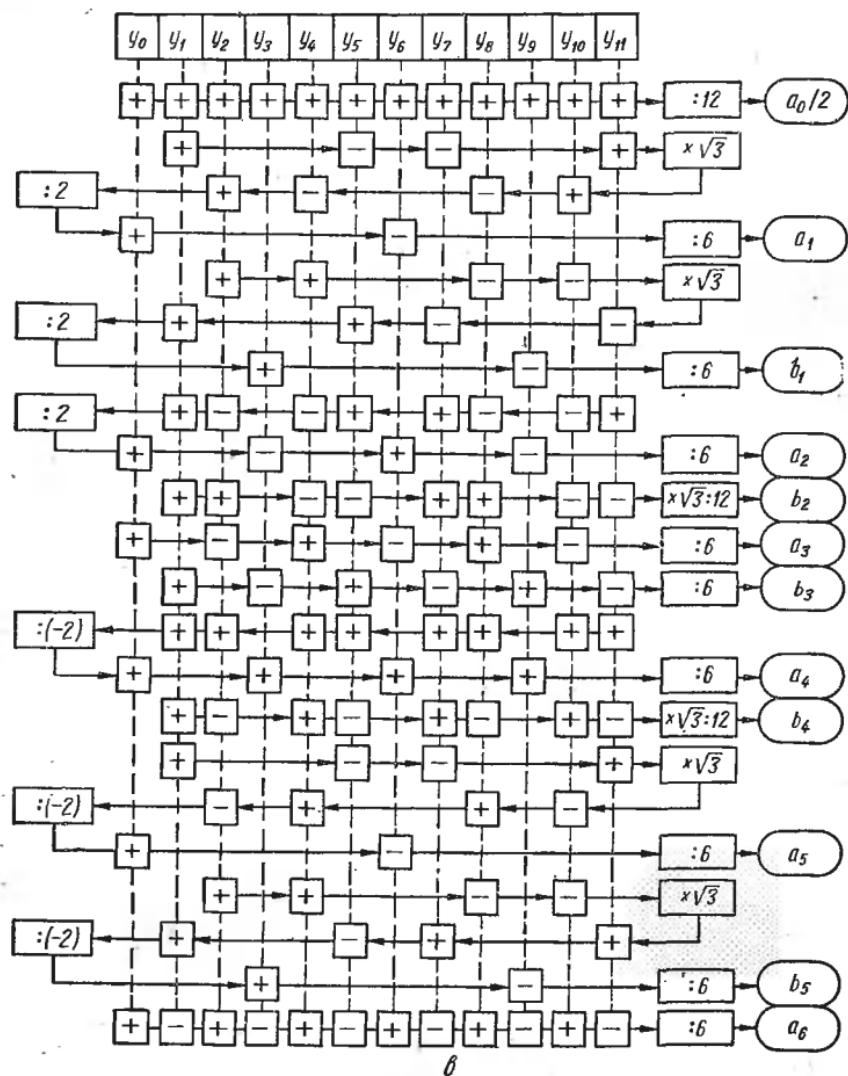
Рис. 16. Схемы вычисления коэффициентов ряда

$$A(v) = \int_{x_a}^{x_b} y(x) \cos vx dx; \quad B(v) = \int_{x_a}^{x_b} y(x) \sin vx dx;$$

$$F(\omega) = \tau \sqrt{A^2(v) + B^2(v)}; \quad \Phi(v) = vx_0 + \operatorname{arctg}(B(v)/A(v));$$

$$F(j\omega) = \tau e^{-j\omega x_0} \int_{x_a}^{x_b} y(x) e^{-j\omega x} dx = \tau (A(v) - jB(v)) e^{-j\omega x_0}.$$

Объем вычислений можно существенно уменьшить при выборе начала отсчета в центре $x_0 = (x_a + x_b)/2$ интервала существования



Фурье по 4 (a), 8 (б), 12 (в) ординатам

функции $y(x)$ и ее представлении суммой четной $y_q(x)$ и нечетной $y_n(x)$ частей. Тогда

$$A(v) = 2 \int_0^{x_b} y_q(x) \cos vx dx; \quad B(v) = 2 \int_0^{x_b} y_n(x) \sin vx dx, \quad (126\text{в})$$

где $y_q(x) = (y(x) + y(-x)) : 2$ и $y_n(x) = (y(x) - y(-x)) : 2$, и общий интервал интегрирования уменьшается в два раза (при $x \geq 0$).

В общем случае подынтегральные выражения в формулах (126) являются быстроосциллирующими и их численное интегрирование чрезвычайно громоздко. Однако трудоемкость вычислений, по крайней мере, для достаточно гладких функций значительно уменьшается при аппроксимации на каждом отрезке Δx не всего подынтегрального выражения, а лишь его огибающей $y_q(x)$ или $y_n(x)$. При этом количество операций резко уменьшается и не зависит от частоты.

Таким образом, при спектральном анализе с помощью микрокалькулятора целесообразно построить графики функций $y_q(x)$ и $y_n(x)$ при $x \geq 0$, выбрать на этих графиках узлы x_i , соответствующие точкам излома функций, и, используя линейную или параболическую аппроксимацию функции $y_q(x)$ и $y_n(x)$ на отрезках Δx , вычислить $A(v)$ и $B(v)$ методами численного интегрирования.

В тех случаях, когда функции $y_q(x)$ и $y_n(x)$ гладкие и не имеют разрывов, использование формул (126) при кусочно-линейной аппроксимации приводит к выражениям

$$A(v) \approx -\frac{2}{v^2} \sum_{i=0}^n \eta_{qi} \cos vx_i; \quad B(v) \approx -\frac{2}{v^2} \sum_{i=0}^n \eta_{ni} \sin vx_i, \quad (127\text{а})$$

где $i = \overline{0, n}$ — номера выбранных узлов аппроксимации x_i , а коэффициенты выражаются через разделенные разности (18а) формулами

$$\eta_i = [x_{i+1}, x_i] - [x_i, x_{i-1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}, \quad (127\text{б})$$

в которых для сохранения вида формул в крайних точках принимают $y_{-1} = y_0$ и $y_{n+1} = y_n$ (значение x_{-1} и x_{n+1} несущественно, но знаменатель должен быть отличен от нуля).

Для определения коэффициентов η_i удобно использовать вычислительный бланк, подобный приведенному в табл. 26. В частном случае постоянного шага $\Delta x = x_{i+1} - x_i = h$ коэффициенты $\eta_i = (y_{i+1} - 2y_i + y_{i-1})/h$.

Формулы (127) дают достаточно точные результаты, если v^2 значительно больше погрешности вычисления суммы ряда в знаменателе.

i	x_i	y_i	$y_i - y_{i-1}$	$x_i - x_{i-1}$	$\Delta y_i / \Delta x_i$	$[x_{i+1}, x_i] - [x_i, x_{i-1}]$
n	x_n	y_n	$y_n - y_{n-1}$	$x_n - x_{n-1}$	$\frac{y_n - y_{n-1}}{x_n - x_{n-1}}$	$0 - \frac{y_n - y_{n-1}}{x_n - x_{n-1}}$
$n-1$	x_{n-1}	y_{n-1}	$y_{n-1} - y_{n-2}$	$x_{n-1} - x_{n-2}$	$\frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}}$	$\frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}} - \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}}$
...
2	x_2	y_2	$y_2 - y_1$	$x_2 - x_1$	$\frac{y_2 - y_1}{x_2 - x_1}$	$\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}$
1	x_1	y_1	$y_1 - y_0$	$x_1 - x_0$	$\frac{y_1 - y_0}{x_1 - x_0}$	$\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}$
0	x_0	y_0	0	1	0	$\frac{y_1 - y_0}{x_1 - x_0} - 0$

Пример 32. Определить вещественную $A(v)$ и мнимую $B(v)$ составляющие спектральной плотности функции (рис. 17, а)

$$y(x) = \begin{cases} 1 - x^2 - \sin \pi x, & |x| \leq 1; \\ 0, & |x| > 1. \end{cases}$$

По аналитическому выражению функции строим графики $y_q(x) = 1 - x^2$ (рис. 17, б) и $y_h(x) = -\sin \pi x$ (рис. 17, в). Так как эти функции достаточно гладкие, целесообразно выбрать постоянный шаг $h = 0,25$ для $y_q(x)$ и $h = 0,2$

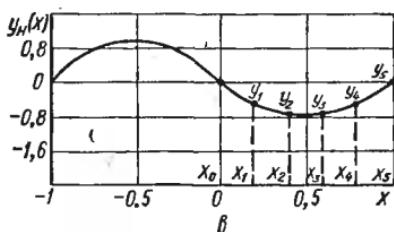
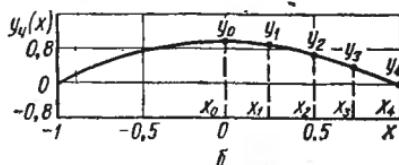
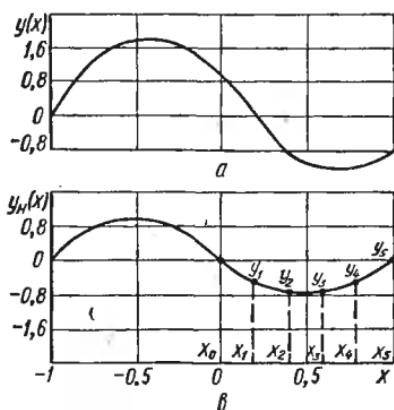


Рис. 17. Графические модели гладкой функции

для $y_h(x)$. Применяя линейно-кусочную аппроксимацию*, находим коэффициенты η_i (табл. 27). Подстановка полученных коэффициентов в формулы (127) дает

$$\begin{aligned} A(v) &\approx (0,5 + \cos 0,25v + \cos 0,5v + \cos 0,75v - 3,5 \cos v)/v^2; \\ B(v) &\approx (-2,24514 \sin 0,2v - 3,63271 \sin 0,4v - 3,63271 \sin 0,6v - \\ &- 2,24514 \sin 0,8v + 5,87785 \sin v)/v^2. \end{aligned}$$

* Для повышения точности значения $y_q(x) = 1; 0,9375; 0,75; 0$ и $y_h(x) = 0; -0,587785; -0,951056; 0$ определены непосредственно по аналитическому выражению заданной функции.

i	x_{q_i}	y_{q_i}	$y_{q_i} - y_{q(i-1)}$	$x_{q_i} - x_{q(i-1)}$	$\Delta y_{q_i} / \Delta x_{q_i}$	η_{q_i}
4	1,0	0	-0,4375	0,25	-1,75	1,75
3	0,75	0,4375	-0,3125	0,25	-1,25	-0,5
2	0,5	0,75	-0,1875	0,25	-0,75	-0,5
1	0,25	0,9375	-0,0625	0,25	-0,25	-0,5
0	0,0	1	0	1	0	-0,25

i	x_{H_i}	y_{H_i}	$y_{H_i} - y_{H(i-1)}$	$x_{H_i} - x_{H(i-1)}$	$\Delta y_{H_i} / \Delta x_{H_i}$	η_{H_i}
5	1,0	0	0,587795	0,2	2,938925	-2,938925
4	0,8	-0,587785	0,363271	0,2	1,816355	1,12257
3	0,6	-0,951056	0	0,2	0	1,816355
2	0,4	-0,951056	0,363271	0,2	-1,816355	1,816355
1	0,2	-0,587785	0,587795	0,2	-2,938925	1,12257
0	0,0	0	0	1	0	-2,938925

Сравнение полученных результатов с точным аналитическим решением показывает, что погрешность расчета не превышает 4% во всем интервале существования функции, кроме точки $v = 4,5$, где относительная погрешность резко возрастает. Однако в этой точке абсолютная погрешность близка к нулю и практически не влияет на результат. Влияние этой погрешности на вычисление $\Phi(v) = \operatorname{arctg} B(v)/A(v)$ можно исключить, не вычисляя фазовый угол в этой точке или положив $B(\hat{v}) = 0$ (или $\Phi(v) = \pm\pi/2$ в зависимости от знака $A(v)$).

При использовании кусочно-параболической аппроксимации (для n четного)

$$\left. \begin{aligned} A(v) &\approx -\frac{2}{v^2} \left(\alpha_{q0} + \sum_{q=2}^{n/2} \left(\alpha_{q0} \cos vx_q - \frac{\beta_{q0}}{v} \sin vx_q \right) \right); \\ B(v) &\approx -\frac{2}{v^2} \left(\frac{\beta_{q0}}{v} + \sum_{q=2}^{n/2} \left(\alpha_{q0} \sin vx_q - \frac{\beta_{q0}}{v} \cos vx_q \right) \right), \end{aligned} \right\} \quad (128a)$$

где $q = 0, 2, 4, \dots$ — номера четных узлов x_i ; $i = 0, 1, 2, 3, \dots$, а коэффициенты

$$\left. \begin{aligned} \alpha_q &= \frac{-y_{q+2} + 4y_{q+1} - 3y_q}{x_{q+2} - x_q} - \frac{3y_q - 4y_{q-1} + y_{q-2}}{x_q - x_{q-2}}; \\ \beta_q &= 4 \left(\frac{y_{q+2} - 2y_{q+1} + y_q}{(x_{q+2} - x_q)^2} - \frac{y_q - 2y_{q-1} + y_{q-2}}{(x_q - x_{q-2})^2} \right) \end{aligned} \right\} \quad (128b)$$

вычисляются по отсчетам для функций $y_n(x)$ и $y_n(x)$ соответственно.

Для сохранения вида формул (128) на краях интервала для $x=0$ и $x=n$ принимают $y_{-2}=y_{-1}=y_0$ и $y_{n+2}=y_{n+1}=y_n$ соответственно.

Если при кусочно-параболической аппроксимации выбрать постоянный шаг $\Delta x = x_{q+2} - x_q = h$, то вычисление коэффициентов

$$\left. \begin{aligned} \alpha_q &= (-y_{q+2} + 4y_{q+1} - 6y_q + 4y_{q-1} - y_{q-2})/h; \\ \beta_q &= 4(y_{q+2} - 2y_{q+1} + 2y_{q-1} - y_{q-2})/h^2 \end{aligned} \right\} \quad (128\text{в})$$

существенно упрощается, хотя это может привести к увеличению числа членов сумм (128а).

Так как формулы (127а) и (128а) содержат в знаменателе величину v^2 , то они дают точные результаты при условии, что v^2 много больше погрешности вычисления суммы тригонометрического ряда в числителях функций.

Пример 33. Найти мнимую составляющую $B(v)$ для заданной в предыдущем примере функции с применением кусочно-параболической аппроксимации.

Разбивая интервал $0 \leq x \leq 1$ функции $y_n(x) = -\sin \pi x$ на два равных сдвоенных интервала с шагом $h = 0,5$, вычисляем значения $y_n(x_i)$ и по формулам (128в) находим $\alpha_2 = (-y_4 + 4y_5 - 6y_2 + 4y_1 - y_0)/h = 0,686288$; $\alpha_4 = (-y_6 + 4y_5 - 6y_4 + 4y_3 - y_2)/h = -3,656856$; $\beta_0 = 4(y_2 - 2y_1 + 2y_{-1} - y_{-2})/h^2 = 6,627424$; $\beta_2 = 4(y_4 - 2y_3 + 2y_1 - y_0)/h^2 = 0$; $\beta_4 = 4(y_6 - 2y_5 + 2y_3 - y_2)/h^2 = -6,627424$.

В соответствии с формулами (128а)

$$B(v) = -2(6,627424(1 - \cos v) : v + 0,686288 \sin 0,5v - 3,656856 \sin v) / v^2.$$

Из сравнения полученного результата с решением предыдущего примера и аналитическим решением видно, что применение кусочно-параболической аппроксимации существенно повышает точность вычислений для главной части ($0,2 \leq v \leq 5,5$) спектра.

Необходимость в аппроксимации отпадает при вычислении спектра кусочно-постоянной функции. В этом случае

$$A(v) = -\frac{2}{v} \sum_{p=0}^m \gamma_{qp} \sin vx_p; \quad B(v) = \frac{2}{v} \sum_{p=0}^m \gamma_{hp} \cos vx_p, \quad (129)$$

где коэффициенты $\gamma_p = y_p'' - y_p'$ определяются разностью (перепадом) значений функции $y_p'' = y_{p+0}$ и $y_p' = y_{p-0}$ у точек $p = \overline{0, m}$ разрыва.

Пример 34. Вычислить $A(v)$ и $B(v)$ для функции, заданной графиком на рис. 18, в.

Раскладывая заданную функцию на четную (рис. 18, б) и нечетную (рис. 18, а) части, вычисляем для $y_n(x)$ коэффициенты $\gamma_{n0} \equiv 0$; $\gamma_{n1} = 2$; $\gamma_{n2} = -7$; $\gamma_{n3} = 4$; $\gamma_{n4} = -2$ и по первой из формул (129) получаем

$$A(v) = (2\sin 2v - 7\sin 6v + 4\sin 9v - 2\sin 11v) 2/(-v).$$

Аналогично определяя для $y_n(x)$ коэффициенты $\gamma_{n1} = 7$; $\gamma_{n2} = -4$; $\gamma_{n3} = 2$; $\gamma_{n4} = -1$, по второй из формул (129) получаем

$$B(v) = (7 - 4 \cos 2v - 4 \cos 5v + 2 \cos 9v - \cos 11v) 2/v.$$

Кусочно-гладкую функцию можно рассматривать как сумму гладкой и кусочно-постоянной функций. Следовательно, формулы для $A(v)$ и $B(v)$ в этом случае будут суммами решений вида (127а) или

(128a) и (129). При этом узлы x_p разрывов функции оказываются граничными точками гладких участков, разбиваемых на отрезки интегрирования. Для того чтобы сохранить общую нумерацию узлов, будем далее в точках разрыва обозначать $y_{i+0} = y_i''$ и $y_{i-0} = y_i'$. Тогда при кусочно-линейной аппроксимации

$$\left. \begin{aligned} A(v) &\approx -\frac{2}{v^2} \sum_{i=0}^n \eta_{qi} \cos vx_i - \frac{2}{v} \sum_{p=0}^m \gamma_{qp} \sin vx_p; \\ B(v) &\approx -\frac{2}{v^2} \sum_{i=0}^n \eta_{hi} \sin vx_i + \frac{2}{v} \sum_{p=0}^m \gamma_{hp} \cos vx_p, \end{aligned} \right\} \quad (130a)$$

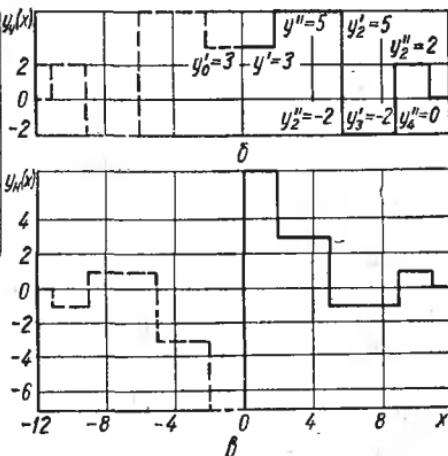
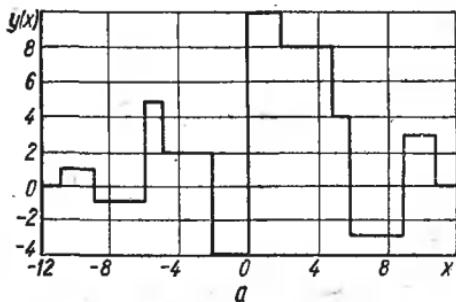


Рис. 18. Графические модели кусочно-постоянной функции

где $\eta_i = [x_{i+1}, x_i] - [x_i, x_{i-1}]$, как в формулах (127б), а $\gamma_p = \overline{y_p} - y_p'$, причем первая сумма берется по всем значениям $i = \overline{0, n}$, а вторая по номерам $p = \overline{0, m}$ точек разрыва функции.

При кусочно-параболической аппроксимации кусочно-гладких функций с конечным числом разрывов

$$\left. \begin{aligned} A(v) &\approx -\frac{2}{v^2} \left(\alpha_{q0} + \sum_{q=2}^{n/2} \left(\alpha_{qg} \cos vx_g - \beta_{qg} \frac{\sin vx_g}{v} \right) \right) - \\ &\quad - \frac{2}{v} \sum_{p=0}^m \gamma_{qp} \sin vx_p; \\ B(v) &\approx -\frac{2}{v^2} \left(\beta_{h0} + \sum_{q=2}^{n/2} \left(\alpha_{hq} \sin vx_g + \beta_{hq} \frac{\cos vx_g}{v} \right) \right) + \\ &\quad + \frac{2}{v} \sum_{p=0}^m \gamma_{hp} \cos vx_p, \end{aligned} \right\} \quad (130б)$$

где коэффициенты α_q и β_q определяются формулами (128б), а $\gamma_p = \overline{y_p} - y_p'$.

Пример 35. Найти $A(v)$ и $B(v)$ для функции

$$y(x) = \begin{cases} e^{-(x+1)} & \text{при } -1 \leq x \leq 0; \\ -e^{-x} & \text{при } 0 \leq x \leq 1; \\ 0 & \text{при } |x| > 1. \end{cases}$$

Раскладывая заданную функцию (рис. 19, а) на четную (рис. 19, б) и нечетную (рис. 19, в) части, видим по графику, что для четной части целесообразно использовать кусочно-линейную, а для нечетной — кусочно-параболическую аппроксимации гладких участков с одним интервалом разбиения.

В соответствии с формулами (130а) находим для четной части
 $\eta_{40} = 0,6321204; \eta_{41} = -0,6321204; \gamma_{41} = (y_1'' - y_1') = -0,3160602$,
откуда

$$A(v) \approx ((v \sin v) : 2 + \cos v - 1) 1,2642408/v^2.$$

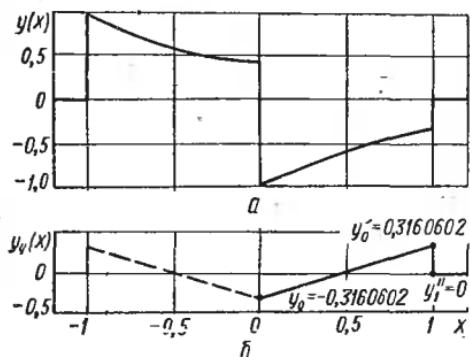
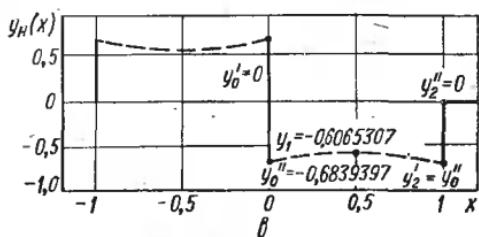


Рис. 19. Графические модели кусочно-непрерывной функции



В соответствии с формулами (130б) для нечетной части находим

$$\alpha_{H2} = \frac{-y_4 + 4y_3 - 3y_2''}{x_4 - x_2} - \frac{3y_2' - 4y_1 + y_0''}{x_2 - x_0} = 0,309636;$$

$$\beta_{H0} = 4 \left(\frac{y_2' - 2y_1 + y_0''}{(x_2 - x_0)^2} \right) = -0,619272; \quad \beta_{H2} = -4 \left(\frac{y_2' - 2y_1 + y_0''}{(x_2 - x_0)^2} \right) = 0,619272; \\ \gamma_{H0} = y_0'' - y_0' = -0,6839397; \quad \gamma_{H2} = y_1'' - y_1' = 0,6839397;$$

$$B(v) \approx \frac{2}{v^2} ((\cos v - 1) (0,6839397v - 0,619272 : v) - 0,309676 \sin v).$$

Сравнение полученных результатов с вычислениями аналитическими методами подтверждает практическое полное их совпадение.

В случае, когда функция $y(x) = y(x + aX)$ имеет период $X = 2\pi/v_1$, ее спектр дискретен и представляется множеством составляющих ряда Фурье

$$C_k = \left[\int_x y(x) e^{-ikv_1 x} dx \right] / X = A_k + jB_k = C_k e^{-j\Phi_k}.$$

В соответствии с преобразованием Фурье (2) связь между спектральной плотностью непериодической функции с конечным интервалом существования и дискретным спектром той же функции, периодически повторяющейся, однозначна и определяется соотношением

$$C_k = F(jkv_1)/X.$$

Следовательно, для спектрального анализа периодических функций достаточно выделить интервал аргумента, равный периоду X функции, и, рассматривая функцию как существующую только в этом интервале, определить одним из рассмотренных методов составляющие $A(v)$ и $B(v)$ спектральной плотности. Разделив их значения на длину интервала X и заменив текущую частоту $v = \omega t$ дискретным множеством $kv_1 = k\omega_1 t$, получим вещественные и мнимые составляющие коэффициентов ряда Фурье

$$A_k = \{a_k\} = 2A(kv_1)/X; \quad B_k = \{b_k\} = 2B(kv_1)/X; \quad k = 0, 1, 2, \dots \quad (131a)$$

или

$$C_k = \{c_k\} = \sqrt{A_k^2 + B_k^2}; \quad \Phi_k = \{\varphi_k\} = \arctg(B_k/A_k); \quad k = 0, 1, 2, \dots \quad (131b)$$

Таким образом, для периодических функций переменные $A(v)$ и $B(v)$ или $C(v)$ и $\Phi(v)$ определяют огибающие дискретных спектров, образованных множествами составляющих (131). Формулы (127) — (131) применимы и по отношению к исходной функции $f(t)$ с началом отсчета t_0 , совпадающим с началом интервала определения функции, не разложенной на четную и нечетную части. В этом случае функцию $f(t)$ аппроксимируют на отрезках разбиения с одинаковыми коэффициентами η_i , α_q , β_q и γ_p для обеих составляющих $A(v)$ и $B(v)$, величины которых в два раза меньше по сравнению со значениями, полученными описанными выше методами.

Пример 36. Определить частотный спектр функции $y(t) = t$ при $0 \leq t \leq 1$ и $y(t) = 0$ при $1 < t < 2$ с периодом $T = 2$.

Выделим один период заданной бесконечной последовательности треугольных импульсов и при $n = 1$ ($t_0 = 0$; $t_1 = 1$; $y_1 = 1$) согласно формуле (127б) вычислим $\eta_0 = 1$; $\eta_1 = -1$; $\gamma_0 = 0$; $\gamma_1 = -1$. С учетом величин составляющих спектральной плотности по формулам (130а) находим

$$A(\omega) = (\omega \sin \omega + \cos \omega - 1)/\omega^2; \quad B(\omega) = (\sin \omega - \omega \cos \omega)/\omega^2.$$

Для $\omega_1 = 2\pi/T = \pi$ при $k = 0, 1, 2, \dots$ согласно формуле (131а) получим

$$a_k = 2A(k\omega_1)/T = 2A(k\pi)/2 = (\cos k\pi - 1)/(k\pi)^2; \\ b_k = 2B(k\omega_1)/T = 2B(k\pi)/2 = -\cos k\pi/k\pi.$$

Раскрыв неопределенность при $k = 0$, определим коэффициент $A(0) = 0,5a_0 = 0,125$. В результате вычислений получим

$$y(t) = 0,25 - 2(0,10132 \cos \omega_1 t + 0,1591549 \sin \omega_1 t - 0,0795774 \sin 2\omega_1 t - 0,0353677 \cos 3\omega_1 t + 0,0530516 \sin 3\omega_1 t + 0,0039789 \sin 4\omega_1 t - 0,0127323 \cos 5\omega_1 t + 0,031309 \sin 5\omega_1 t - \dots).$$

Однако спектральный анализ непосредственно по заданной функции целесообразен лишь для простейших функций, подобных рассмотренной в последнем примере. В общем случае для уменьшения объема вычислений следует использовать методику спектрального анализа, основанную на разложении исходной функции на четную и нечетную составляющие.

8. РЕШЕНИЕ ИНЖЕНЕРНО-ЭКОНОМИЧЕСКИХ ЗАДАЧ

В процессе инженерного проектирования, особенно на этапе экономического обоснования проекта, возникают специфические задачи, связанные в основном с вычислением процентов и планированием производства.

Процентом и промилле называют соответственно одну сотую ($0,01a = 1\% \text{ от } a$) и одну тысячную ($0,001a = 1\text{\%}_0 \text{ от } a$) части числа. Большинство задач на вычисление процентов сводятся к использованию пропорции $x : y = p\% : 100\%$, где x — число, выражаемое в процентах $p\% = p(x : y)\%$ относительно числа y , принимаемого за 100%. Например, если заданы числа $a = 36,2$ и $b = 22,1$, то число a составляет $p(a : b)\% = 36,2 \cdot 100 : 22,1 = 163,8\%$ от числа b , а число b составляет $p(b : a)\% = 22,1 \cdot 100 : 36,2 = 61,05\%$ от числа a .

Иногда приходится вычислять проценты от процентов (сложные проценты) или определять числа по сложным процентам. Обычно подобные вычисления связаны с изменением исходного числа на некоторую величину за определенный отрезок времени. Если через равные отрезки времени исходное число x_0 изменяется на Δx , то за n таких отрезков переменная $x = x_0(1 \pm \Delta x)^n$, что соответствует изменению исходного числа на $\Delta p\% = [(1 + \Delta x)^n - 1] \cdot 100\%$. Например, если стоимость оборудования $x = 10\ 000$ р. ежегодно уменьшается за счет амортизации на 7,5%, то через 6 лет его стоимость составляет $p\% = (1 - 0,075)^6 \cdot 100\% = 62,64\%$ от исходной или $x = x_0p = 10\ 000 \cdot 0,6264 = 6240$ р., т. е. уменьшается на $\Delta p\% = 100 - 62,64 = 37,36\%$.

Аналогичные задачи возникают при вычислении сложных процентов для начислений на фонд заработной платы, банковские кредиты, вклады в сберкассах и т. п. Например, для срочного вклада $x_0 = 100$ р. при 3% годовых начислений через 7 лет сумма вклада возрастет до $p\% = (1 + 0,03)^7 \cdot 100 = 122,9873$ или $x = x_0p \approx 122,98$ р.

Входные языки микрокалькуляторов некоторых типов содержат оператор вычисления процентов. В этом случае при нажатии соответствующей клавиши после деления или умножения двух чисел результат соответственно умножается или делится на 100. В специализированных микрокалькуляторах, предназначенных для бухгалтерских и экономических расчетов (например, типа «Электроника Б3-26К») предусмотрено выполнение и более сложных вычислений с процентами.

Задачи производственно-экономического планирования обычно содержат большое число переменных и для их решения используют ЭВМ с достаточно большой емкостью запоминающих устройств. Однако в инженерной практике встречаются относительно простые задачи планирования производства, которые можно успешно решать при помощи микрокалькуляторов.

Одна из основных задач планирования производства формулируется следующим образом [11]. Для производства продукта ис-

пользуется n различных технологий и m различных видов ресурсов (сырья, полуфабрикатов, электроэнергии, фонда зарплаты и т. п.), причем по j -й технологии ($j = 1, 2, \dots, n$) в единицу времени расходуется a_{ij} единиц i -го вида ресурсов ($i = 1, 2, \dots, m$) с запасами a_i и производится p_j единиц продукта. Если x_j — время производства по j -й технологии, то при плане $x = (x_1, \dots, x_n)$ будет произведено $\Phi = p_1x_1 + \dots + p_nx_n$ единиц продукта и израсходовано $a_{11}x_1 + \dots + a_{nn}x_n \leq a_i$ единиц i -го вида ресурсов.

Приняв за единицу стоимости ресурсов стоимость единицы выпускаемого продукта, можно выразить в этом масштабе цен стоимость всего запаса ресурсов $z = a_1u_1 + \dots + a_mu_m$ при стоимостях u_i каждого вида ресурсов. В этом случае стоимость затрат в единицу времени на производство по j -й технологии $a_{1j}u_1 + \dots + a_{mj}u_m$ не может быть меньше стоимости q_j продукта, изготовленного по этой технологии. Чем меньше непроизводительных затрат, тем ближе затраты к стоимости продукта и при рентабельном плане стоимость z всех затрат производства равна стоимости Φ произведенного продукта.

Таким образом, задача планирования производства для приведенных условий математически формулируется как двойственная задача линейного программирования: минимизировать функцию

$$z = a_1u_1 + \dots + a_mu_m \quad (132a)$$

при ограничениях

$$a_{1j}u_1 + \dots + a_{mj}u_j \geq q_j; \quad j = 1, 2, \dots, n$$

или максимизировать функцию

$$\Phi = p_1x_1 + \dots + p_nx_n \quad (132b)$$

при ограничениях

$$a_{ii}x_1 + \dots + a_{nn}x_n \leq a_i; \quad i = 1, 2, \dots, m.$$

Эта двойственная задача решается симплекс-методом одновременно [11], так как оптимальный план, соответствующий максимальному выпуску продукции при заданных запасах ресурсов, удовлетворяет и требованию максимальной рентабельности производства. Рассмотренным ранее симплекс-методом решают и многие другие инженерно-экономические задачи.

При вычислениях по этому методу после каждого жорданова исключения заполняют новый вычислительный бланк (симплекс-таблицу). Вычисления элементов новой симплекс-таблицы с помощью микрокалькуляторов согласно формуле (100в) удобно выполнять по следующему алгоритму:

1. Для выбранного разрешающего элемента $\alpha_{pq}^{(k-1)}$ вычисляют значение

$$\alpha_{pq}^{(k)} = 1/\alpha_{pq}^{(k-1)}, \quad (133a)$$

которое записывают в новую таблицу.

2. Остальные элементы p -й строки новой таблицы вычисляют по формуле

$$\alpha_{pj}^{(k)} = \alpha_{pj}^{(k-1)} / \alpha_{pq}^{(k-1)}, \quad (133б)$$

реализуемой на микрокалькуляторе при помощи операций с константой.

3. Остальные элементы q -го столбца новой таблицы вычисляют по формуле

$$\alpha_{iq}^{(k)} = -\alpha_{iq}^{(k-1)} / \alpha_{pq}^{(k-1)}. \quad (133в)$$

4. Элементы остальных строк и столбцов новой таблицы вычисляют по формулам

$$\alpha_{ij}^{(k)} = \alpha_{ij}^{(k-1)} - \alpha_{pj}^{(k-1)} \alpha_{iq}^{(k-1)} / \alpha_{pq}^{(k-1)}. \quad (133г)$$

Рассмотрим некоторые частные задачи производственного планирования, решаемые с помощью симплекс-метода. Обычно таким задачам дают конкретные названия, но по их образцу решают и многие другие задачи, имеющие аналогичную математическую формулировку.

Задача о *максимальном выигрыше* производства n видов изделий, каждое из которых дает выигрыш (например, прибыль) p_j условных единиц на единицу продукции из m видов ресурсов, формулируется как задача максимизации функции (132 б), где a_{ij} — расход i -го вида ресурсов ($i = 1, 2, \dots, m$) на изготовление одного изделия j -го вида, а x_j — количество изделий j -го вида ($j = 1, 2, \dots, n$).

Пример 37. Производство изделий двух видов дает выигрыш в условиях единицах $p_1 = 3$ и $p_2 = 4$ соответственно на единицу изделия каждого типа. При плановом задании по номенклатуре изделий $x_1 \geq 10; x_2 \geq 10$ и запасах (в единицах стоимости) сырья $a_1 = 1000$, электроэнергии $a_2 = 500$, фонда зарплаты $a_3 = 400$ необходимо определить оптимальное количество x_1 и x_2 изделий, если расход ресурсов на их изготовление определяется неравенствами

$$20x_1 + 10x_2 \leq a_1; \quad 10x_1 + 20x_2 \leq a_2; \quad 10x_1 + 10x_2 \leq a_3.$$

Опишем исходные данные задачи симплекс-таблицей, приведенной в нулевой части табл. 28. Выбирая в соответствии с описанными в предыдущей главе алгоритмами разрешающий элемент $\alpha_{41} = -1$, по формулам (133) вычисляем значения элементов новой симплекс-таблицы, приведенной в первой части табл. 28. Выбирая в этой таблице разрешающий элемент $\alpha_{52} = -1$, получаем новую таблицу (вторая часть табл. 28), которая отображает опорное решение, так как все свободные члены неотрицательны.

Выбирая в полученной таблице разрешающий элемент $\alpha_{22} = 20$ для устранения отрицательного элемента в Ф-строке, вычисляем по формулам (133) элементы таблицы, приведенной в третьей части табл. 28. Выбирая разрешающий элемент $\alpha_{81} = 5$ и вычисляя только свободные члены и элементы Ф-строки, получаем симплекс-таблицу (четвертая часть табл. 28), отображающую оптимальное решение. Следовательно, максимальный выигрыш составляет $\max \Phi = 130$ условных единиц при количестве изделий $x_1 = 30$ и $x_2 = 10$, полном использовании запасов электроэнергии и фонда зарплаты ($g_2 = g_3 = 0$) и сохранении $g_1 = 300$ условных единиц сырья. Переменные $g_4 = 20$ и $g_5 = 0$ характеризуют перевыполнения планового задания для обоих видов изделий.

0		$-x_1$	$-x_2$	1	3		$-g_4$	$-g_2$	1
	g_1	20	10	1000		g_1	15	-0,5	600
	g_2	10	20	500		g_5	0,5	0,05	10
	g_3	10	10	400		g_3	5	-0,5	100
	g_4	-1	0	-10		x_1	-1	0	10
	g_5	0	-1	-10		x_2	0,5	0,05	20
	Φ	-3	-4	0		Φ	-1	0,2	110
1		$-g_4$	$-x_2$	1	4		$-g_3$	$-g_2$	1
	g_1	20	10	800		g_1			300
	g_2	10	20	400		g_5			0
	g_3	10	10	300		g_4			20
	x_1	-1	0	10		x_1			30
	g_5	0	-1	-10		x_2			10
	Φ	-3	-4	30		Φ	0,2	0,3	130
2		$-g_4$	$-g_5$	1					
	g_1	20	10	700					
	g_2	10	20	200					
	g_3	10	10	200					
	x_1	-1	0	10					
	x_2	0	-1	10					
	Φ	-3	-4	70					

Рассмотренную методику используют также при решении задач об оптимальных смесях, сводящихся к определению оптимального числа x_i изделия каждого вида ($i = 1, 2, \dots, n$), минимизирующего функцию

$$z = p_1 x_1 + \dots + p_n x_n,$$

где p_i — вес (цена, стоимость, количественный показатель качества и т. п.) каждого изделия, содержащего a_{ij} компонентов вида $j = 1, \dots, m$, причем во всех изделиях должно содержаться не менее a_j компонентов каждого вида.

Так как условия задачи отображаются неравенствами

$$a_{1j} x_1 + \dots + a_{nj} x_n \geq a_j,$$

то такая задача сводится к предыдущей при $z = -\Phi$.

Более сложны общие задачи планирования производства. Пусть имеется n технологий ($j = 1, 2, \dots, n$) для выпуска комплектных изделий, состоящих из компонентов l_k при $k = 1, 2, \dots, s$. Для их изготовления используют m видов ресурсов с запасами a_i ресурса каждого вида ($i = 1, 2, \dots, m$), причем по j -й техно-

логии i -й вид ресурсов используется в количестве a_{ii} единиц и соответствует изготовлению b_k единиц k -го компонента. План $x = (x_1, \dots, x_n)$ составляется из числа x_j циклов производства по j -й технологии.

Так как в каждом комплекте изделия содержится l_k единиц каждого компонента, то из последних можно изготовить $(b_{k1}x_1 + \dots + b_{kn}x_n)/l_k$ комплектов, причем число полных комплектов равно наименьшему из этих частных. Задача заключается в выборе такого плана, при котором из имеющихся ресурсов изготавливается максимальное число полных комплектов изделия.

Таким образом, для неотрицательных решений системы неравенств $a_{j1}x_1 + \dots + a_{jn}x_n \leq a_j$ необходимо найти максимальное число полных комплектов при минимальном числе компонентов каждого вида

$$\max \min (b_{k1}x_1 + \dots + b_{kn}x_n)/l_k.$$

При $s = 1$ задача сводится к минимизации функции

$$z = (b_{11}x_1 + \dots + b_{1n}x_n)/l_1.$$

В общем случае $s > 1$ рассматриваемая задача введением дополнительной неотрицательной переменной ξ сводится к минимизации $\Phi = \xi$ при ограничениях

$$\begin{aligned} a_{j1}x_1 + \dots + a_{jn}x_n &\leq a_j; \quad j = 1, 2, \dots, m; \\ (b_{k1}x_1 + \dots + b_{kn}x_n)/l_k &\geq \xi; \quad k = 1, 2, \dots, s; \\ x_j &\geq 0; \quad \xi \geq 0; \quad j = 1, 2, \dots, n. \end{aligned}$$

Упрощенный вариант рассматриваемой задачи формулируют следующим образом. Изделия собирают из l_k компонентов каждого вида $k = 1, 2, \dots, s$, изготовленных на n станках, причем имеется c_j станков типов $j = 1, 2, \dots, n$ на каждом из которых можно изготовить a_{kj} деталей k -го типа. Необходимо так распределить производственные задания между станками, чтобы обеспечить максимальный выход комплектов изделия. В этой формулировке задача сводится к отысканию $\max \Phi = \max \xi$ при ограничениях

$$\begin{aligned} x_{1j} + \dots + x_{sj} &= c_j; \quad a_{j1}x_{1j} + \dots + a_{js}x_{sj} \geq \xi; \quad j = 1, 2, \dots, n; \\ k = 1, 2, \dots, s; \quad x_{kj} &\geq 0; \quad \xi \geq 0 \end{aligned}$$

и отличается от предыдущей лишь системой равенств.

Пример 38. Изделие собирают из двух деталей первого типа ($k = 1$) и одной второго типа ($k = 2$), изготовленных на станках двух типов ($j = 1, 2$), число и производительность которых указаны в табл. 29. Необходимо выбрать числа x_{kj} станков каждого типа для изготовления определенного типа деталей так, чтобы выпуск комплектных изделий был максимальным.

Обозначив буквой ξ число полных комплектов изделия, опишем условия задачи выражениями

$$\begin{aligned} x_{11} + x_{21} &= 10; \quad x_{12} + x_{22} = 5; \quad \Phi = \xi; \\ (50x_{11} + 400x_{12})/2 &\geq \xi; \quad 100x_{21} + 200x_{22} \geq \xi, \end{aligned}$$

и, преобразовав их к виду

$$0 = -x_{11} - x_{21} + 10; \quad 0 = -x_{12} - x_{22} + 5; \quad \Phi = -(-\xi);$$

$$g_1 = 50x_{11} + 400x_{12} - 2\xi; \quad g_2 = 100x_{21} + 200x_{22} - \xi,$$

составим исходную симплекс-таблицу, приведенную в нулевой части табл. 30.

В первом столбце составленной таблицы элементу $\alpha_{11} = 1$ соответствует наименьшее отношение $a_1/\alpha_{11} \geq 0$ и его можно выбрать в качестве разрешающего элемента. Вычислив по формулам (133) элементы новой симплекс-таблицы (первая часть табл. 30), аналогично выбираем в качестве разрешающего ее элемент $\alpha_{22} = 1$ и получаем симплекс-таблицу (вторая часть табл. 30), соответствующую опорному решению.

Выбрав в последней из таблиц разрешающий элемент $\alpha_{44} = 1$, переходим к таблице, приведенной в нулевой части табл. 31. Выбрав в этой таблице разрешающий элемент $\alpha_{31} = 250$, получим симплекс-таблицу (первая часть табл. 31), соответствующую оптимальному решению (неотрицательны все элементы столбца свободных членов и Φ -строки).

Таблица 29

Номер детали	Тип станка		Число деталей в изделии
	1	2	
1	50	400	2
2	100	200	1
Число станков	10	5	

Таблица 30

0		$-x_{11}$	$-x_{12}$	$-x_{21}$	$-x_{22}$	$-\xi$	Φ
	0	1	0	1	0	0	10
	0	0	1	0	1	0	5
	g_1	-50	-400	0	0	2	0
	g_2	0	0	-100	-200	1	0
	Φ	0	0	0	0	-1	0
1			$-x_{12}$	$-x_{21}$	$-x_{22}$	$-\xi$	0
	x_{11}		0	1	0	0	10
	0		1	0	1	0	5
	g_1		-400	50	0	2	500
	g_2		0	-100	-200	1	0
	Φ		0	0	0	-1	0
2				$-x_{21}$	$-x_{22}$	$-\xi$	1
	x_{11}			1	0	0	10
	x_{12}			0	1	0	5
	g_1			50	400	2	2500
	g_2			-100	-200	1	0
	Φ			0	0	-1	0

0		x_{21}	x_{22}	g_2	1	1		g_1	x_{22}	g_2	1
x_{11}		1	0	0	10	x_{11}	-0,004	3,2	-0,08	0	
x_{12}		0	1	0	5	x_{12}	0	1	0	5	
g_1		250	800	-2	2500	x_{21}	1	3,2	-0,08	10	
ξ		-100	-200	1	0	ξ	0,4	120	0,2	1000	
Φ		-100	-200	1	0	Φ	0,4	120	0,2	1000	

Многие инженерные задачи аналогичны задаче о минимизации отходов. Она заключается в таком разбиении n типов полуфабрикатов по a_i штук на заготовки деталей, при котором число комплектов заготовок максимально при минимальном количестве использованного полуфабриката.

Если обозначить x_{ij} число заготовок i -го типа, которые намечено разбить j -м способом ($j = \overline{1, m}$), то из них будет получено $a_{ijk}x_{ij}$ деталей k -го типа ($k = \overline{1, s}$) и из всех n типов полуфабрикатов число полученных деталей

$$b_s = \sum_{i=1}^n \sum_{j=1}^m a_{ijk}x_{ij}.$$

Если из этого числа можно собрать b_s/l_k комплектных изделий, то задача сводится к максимизации $\Phi = \xi$ при ограничениях

$$\frac{1}{l_k} \sum_{i=1}^n a_{ijk}x_{ij} \geq \xi; \quad \sum_{j=1}^m x_{ij} = a_i; \quad k = \overline{1, s}; \quad i = \overline{1, n}; \quad x_{ij} \geq 0; \quad \xi \geq 0.$$

Пример 39. Из двух типов прутьев (50 шт. длиной 5 м каждый и 80 шт. длиной 3,5 м) изготавливают комплекты, содержащие два прута длиной 2 м и один длиной 1,5 м. Как разрезать имеющиеся прутья для получения наибольшего числа полных комплектов заготовок?

Прут длиной 5 м можно разрезать на заготовки требуемой длины следующими способами: 1) на два прута длиной 2 м; 2) на три прута длиной 1,5 м; 3) на один прут длиной 2 м и два прута длиной 1,5 м. Прут длиной 3,5 м можно разрезать на заготовки следующими способами: 1) один прут длиной 2 м и один длиной 1,5; 2) два прута длиной 1,5 м. Остальные варианты отбрасываем в связи с явно большими отходами, но их также можно учесть.

Обозначим x_{1j} ($j = 1, 2, 3$) прутья длиной 5 м, разрезанные одним из перечисленных трех способов, и x_{2j} ($j = 1, 2$) прутья длиной 3,5 м, разрезанные одним из двух способов. Тогда рассматриваемая задача сводится к максимизации числа комплектов заготовок $\Phi = \xi$ при ограничениях

$$x_{11} + x_{12} + x_{13} = 50; \quad x_{21} + x_{22} = 80; \quad 0,5(2x_{11} + x_{13} + x_{21}) \geq \xi; \\ 3x_{12} + 2x_{13} + x_{21} + 2x_{22} \geq \xi.$$

Составим по условиям задачи табл. 32 и после исключения О-строк и ρ_2 получим симплекс-таблицу, приведенную в первой части табл. 33 (ξ -строку исключаем, так как она совпадает с Φ -строкой). Выбирая для наибольшего

Таблица 32

	$-x_{11}$	$-x_{12}$	$-x_{13}$	$-x_{21}$	$-x_{22}$	ξ	1
0	1	1	1	0	0	0	50
0	0	0	0	1	1	0	80
g_1	-2	0	-2	-1	0	2	0
g_2	0	-3	-2	-1	-2	1	0
Φ	0	0	0	0	0	-1	0

по модулю элемента Φ -строки разрешающий элемент $\alpha_{31} = 8$, вычислим элементы новой симплекс-таблицы, приведенной во второй части табл. 33. Выбрав для единственного отрицательного элемента Φ -строки разрешающий элемент $\alpha_{32} = 0,625$, получим третью часть табл. 33. Следовательно, наибольшее число $\max \Phi = \max \xi = 88$ комплектов заготовок получим при разрезании прутьев длиной 5 м способами x_{11} и x_{13} и прутьев длиной 3,5 м способом x_{21} .

Таблица 33

1		$-x_{12}$	$-x_{13}$	$-x_{22}$	$-g_2$	1
	x_{11}	1	1	0	0	50
	x_{21}	0	0	1	0	80
	g_1	8	5	3	-2	20
	Φ	-3	-2	-1	1	80
2		g_1	$-x_{13}$	$-x_{22}$	$-g_2$	1
	x_{11}	-0,125	0,375	-0,375	0,25	47,5
	x_{21}	0	0	1	0	80
	x_{12}	0,125	0,625	0,375	-0,25	2,5
	Φ	0,375	-0,125	0,125	0,25	87,5
3		$-g_1$	$-x_{12}$	$-x_{22}$	$-g_2$	1
	x_{11}					46
	x_{21}					80
	x_{13}					4
	Φ	0,4	0,2	0,2	0,2	88

Распределительная задача сводится к такому распределению k_j объектов n типов ($j = 1, 2, \dots, n$) между рабочими местами с номерами $i = 1, 2, \dots, m$, для которого минимальны суммарные затраты при заданном минимальном объеме производства a_i на i -м месте, и

производством j -м типом объекта объема a_{ij} продукции при затратах b_{ij} . Эта задача сводится к минимизации функции

$$z = \sum_{j=1}^n \sum_{i=1}^m b_{ij} x_{ij}$$

при ограничениях

$$a_{i1}x_{i1} + \dots + a_{in}x_{in} \geq a_i; \quad x_{i1} + \dots + x_{nj} = k_j,$$

где x_{ij} — число объектов j -го типа, закрепленных за i -м местом.

Пример 40. Станки трех типов ($j = 1, 2, 3$) распределяются между двумя цехами ($i = 1, 2$) при указанных в табл. 34 объеме производства (первое число) и затратах (второе число). При заданном плане производства не менее 300 единиц продукции для первого цеха и 200 для второго необходимо так распределить станки различных типов между цехами, чтобы выполнить и, по возможности, перевыполнить план при наименьших затратах.

Введя две дополнительные неотрицательные переменные g_1 и g_2 , отобразим условия задачи выражениями

$$g_1 = 10x_{11} + 20x_{12} + 20x_{13} \geq 300;$$

$$g_2 = 10x_{21} + 20x_{22} + 10x_{23} \geq 200;$$

$$x_{11} + x_{21} = 40; \quad x_{12} + x_{22} = 15; \quad x_{13} + x_{23} = 20;$$

$$z = 10x_{11} + 50x_{12} + 20x_{13} + 15x_{21} + 30x_{22} + 5x_{23}$$

и составим исходную симплекс-таблицу, приведенную в нулевой части табл. 35. Последовательно устранив О-строки выбором разрешающих элементов $\alpha_{31} = 1$, $\alpha_{44} = 1$, $\alpha_{54} = 1$, получим симплекс-таблицу, приведенную в первой части табл. 35. Эта таблица соответствует не только опорному (все свободные члены положительны), но и оптимальному (все элементы з-строки отрицательны) решению, причем минимальным расходам труда $z = 950$ соответствует передача всех станков первого типа первому цеху ($x_{11} = 40$) и остальных ($x_{21} = 15$, $x_{23} = 20$) второму цеху. При этом первый цех выпустит 400 и второй цех 500 условных единиц продукции сверх плана.

Таблица 34

Номер цеха	Тип станка			Минимальный план производства
	1	2	3	
1	10 10	20 50	20 20	300
2	10 15	20 30	10 5	200
Число станков	40	15	20	

В задаче о назначениях, близкой к варианту предыдущей задачи, n объектов распределяют на m видов работ так, чтобы каждый объект выполнял только один вид работы при максимальной суммарной производительности.

Если обозначить c_{ij} производительность i -го объекта на j -й работе ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$), то рассматриваемая задача сводится к такому выбору n элементов матрицы c_{ij} , взятых по одному от каждого столбца и каждой строки, при котором их сумма минимальна. Обозначив $x_{ij} = 1$, если i -й объект

Таблица 35

0		$-x_{11}$	$-x_{12}$	$-x_{13}$	$-x_{21}$	$-x_{22}$	$-x_{23}$	1
	g_1	-10	-20	-20	0	0	0	300
	g_2	0	0	0	-10	-20	-10	200
	0	1	0	0	1	0	0	40
	0	0	1	0	0	1	0	15
	0	0	0	1	0	0	1	20
	z	-10	-50	-20	-15	-30	-5	0
1		0	$-x_{12}$	$-x_{13}$	$-x_{21}$	0	0	1
	g_1			-20	-20	10		700
	g_2			20	10	-10		700
	x_{11}			0	0	1		40
	x_{22}			1	0	0		15
	x_{23}			0	1	0		20
	z			-20	-15	-5		950

назначен на j -й вид работы, и $x_{ij} = 0$ в противном случае, задачу сводят к максимизации суммарного произведения

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

при ограничениях

$$\begin{aligned} x_{i1} + x_{i2} + \dots + x_{in} &= 1; \quad i = 1, 2, \dots, n; \\ x_{1j} + x_{2j} + \dots + x_{nj} &= 1; \quad j = 1, 2, \dots, n, \end{aligned}$$

соответствующих выполнению одним объектом работы одного вида и каждого вида работы одним объектом.

Таблица 36

Номер станка	Виды работ		
	1	2	3
1	2	3	1
2	1	0	3
3	1	5	0

Пример 41. Производительность c_{ij} (в условных единицах) трех станков при выполнении трех видов работ описывается данными табл. 36. Требуется так распределить задания x_{ij} станкам i на выполнение работ j , чтобы суммарная производительность была максимальной.

В рассматриваемом случае суммарная производительность станков описывается равенством

$$\Phi = 2x_{11} + 3x_{12} + x_{13} + x_{21} + 3x_{23} + x_{31} + 5x_{32}$$

при ограничениях

$$\begin{aligned} x_{11} + x_{12} + x_{13} &= 1; \quad x_{21} + x_{22} + x_{23} = 1; \quad x_{31} + x_{32} + x_{33} = 1; \\ x_{11} + x_{21} + x_{31} &= 1; \quad x_{12} + x_{22} + x_{32} = 1; \quad x_{13} + x_{23} + x_{33} = 1. \end{aligned}$$

Составив по этим условиям табл. 37, после устранения нуль-строк и исключения строки, содержащей только нулевые элементы, получим первую часть табл. 38. Выбрав для устранения отрицательного свободного члена разрешающий элемент $\alpha_{22} = 1$, получим вторую часть табл. 38. Выбрав в этой симп-

Таблица 37

	$-x_{11}$	$-x_{12}$	$-x_{13}$	$-x_{21}$	$-x_{22}$	$-x_{23}$	$-x_{31}$	$-x_{32}$	$-x_{33}$	1
0	1	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	0	0	0	0	1
0	0	0	0	0	0	0	1	1	1	1
0	1	0	0	1	0	0	1	0	0	1
0	0	1	0	0	1	0	0	1	0	1
0	0	0	1	0	0	1	0	0	1	1
Φ	-2	-3	-1	-1	0	-3	-1	-5	0	0

Таблица 38

1		$-x_{22}$	$-x_{23}$	$-x_{32}$	$-x_{33}$	1
	x_{11}	-1	-1	-1	-1	-1
	x_{21}	1	1	0	0	1
	x_{31}	0	0	1	1	1
	x_{12}	1	0	1	0	1
	x_{13}	0	1	0	1	1
	Φ	0	-3	-3	0	4

2		$-x_{22}$	$-x_{21}$	$-x_{32}$	$-x_{33}$	1
	x_{11}	0	1	-1	-1	0
	x_{23}	1	1	0	0	1
	x_{31}	0	0	1	1	1
	x_{12}	1	0	1	0	1
	x_{13}	-1	-1	0	1	0
	Φ	3	3	-3	0	7

3		$-x_{22}$	$-x_{21}$	$-x_{31}$	$-x_{33}$	1
	x_{11}					1
	x_{23}					1
	x_{32}					1
	x_{12}					0
	x_{13}					0
	Φ	3	3	3	3	10

лекс-таблице разрешающий элемент $\alpha_{33} = 1$ для устранения последнего отрицательного элемента Φ -строки, получим третью часть табл. 38. Последний столбец этой части показывает, что максимальная производительность достигается при распределении заданий 1,2 и 3 по станкам 1,3 и 2 соответственно.

Симплекс-метод достаточно универсален, но решение ряда задач планирования производства, сводящихся к задачам линейного программирования, упрощается при использовании других методов. В частности, простейшие варианты задач о назначениях, подобные рассмотренному в примере 41, удобно решать методом перебора с предварительным определением допустимых сочетаний (по одному от каждой строки и столбца) элементов исходной таблицы (матрицы) в соответствии с формулой (96) и реализующим ее алгоритмам. Определяя, например, согласно этой формуле контурное число, столбцы которого содержат индексы элементов для допустимых сочетаний, по табл. 36 получим

$$\beta = \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix}_{\text{mod } 2} = \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 3 \\ 2 & 3 & 1 & 3 & 1 & 2 \\ 3 & 2 & 3 & 1 & 2 & 1 \end{bmatrix}.$$

Вычисляя для каждого столбца суммы производительности станков с соответствующими индексами, найдем, что наибольшая производительность в 10 условных единиц соответствует второму столбцу контурного числа при назначениях x_{11} , x_{23} и x_{32} , что совпадает с результатами вычислений в примере 41.

Глава 3

ПРОГРАММИРОВАНИЕ МИКРОКАЛЬКУЛЯТОРОВ

1. НАЧАЛА ПРОГРАММИРОВАНИЯ

Электронные цифровые вычислительные машины различных типов, включая микро-ЭВМ типа программируемых микрокалькуляторов, автоматически решают математические задачи, последовательно выполняя операции над числами в соответствии с заранее составленной и введенной в память ЭВМ программой вычислений.

Программа автоматических вычислений представляет собой изложение выбранного способа решения задачи на языке, «понятном» ЭВМ используемого типа. Поэтому перед составлением программы и ее вводом в запоминающие устройства машины выбранный способ решения задачи отображают алгоритмом — четким и однозначным описанием последовательности операций над исходными числами, приводящей к искомому результату. Таким образом, программирование сложных задач в общем случае включает три взаимосвязанных этапа:

1. Выбор математической формулировки условий задачи, упрощающей ее решение.

2. Выбор алгоритма, обеспечивающего решение задачи с заданной точностью при минимальных затратах времени.

3. Составление программы с рациональным использованием лексических и грамматических особенностей входного языка, а также емкости запоминающих устройств ЭВМ используемого типа.

Программы ЭВМ первых поколений составляли на языках машинных кодов. В таких языках каждый шаг (команда) программы образован в определенной системе счисления множеством цифровых кодов, отображающих адреса регистров или ячеек памяти, в которых хранятся операнды и куда заносятся результаты операции, а также кодов выполняемой и следующей операций. Подобные программы громоздки, а для их составления, отладки и проверки требуются значительные затраты времени.

Для программирования современных ЭВМ в основном используют алгоритмические языки, словами которых являются символы

операторов. Символами операторов могут служить буквы, цифры, специальные знаки, а также их определенные сочетания, например, русские и английские слова. Последовательность операторов образует программу.

Введенная в ЭВМ программа, составленная на входном алгоритмическом языке, с помощью специального устройства и вспомогательной программы (транслятора) автоматически переводится на внутренний язык машинных кодов. При этом каждый оператор входного языка (реализуемый, как правило, некоторым множеством команд на языке машинных кодов) фактически пускает внутреннюю программу (микропрограмму) его реализации, сложность которой зависит от вида оператора. Число подобных операторов и сложность микропрограмм определяет уровень входного алгоритмического языка.

Входные языки программируемых микрокалькуляторов являются алгоритмическими языками более высокого уровня, чем входные языки универсальных ЭВМ. Это существенно упрощает программирование микрокалькуляторов пользователями, не имеющими специальной подготовки.

Программируемые микрокалькуляторы работают как в обычном режиме ввода операндов и выполнения операций последовательным нажатием клавиш, так и в режиме автоматических вычислений по заданной программе. В последнем случае программой называют описывающее алгоритм решения задачи предложение на входном языке микрокалькулятора, которое предварительно вводится в запоминающее устройство и автоматически выполняется по специальной команде после ввода исходных данных.

В большинстве программируемых микрокалькуляторов программа вводится нажатием тех же клавиш, что и при вычислениях в обычном режиме, причем она стирается при вводе новой программы или выключении микрокалькулятора. Поэтому программируемый режим целесообразно использовать лишь в тех случаях, когда вычисления по введенной программе выполняются многократно, например, при изменении исходных данных или итерационных алгоритмах.

Удобство записи программы на алгоритмическом языке связано также с ее аналогией обычному словесному или словесно-формульному описанию алгоритма решения задачи по пунктам (блокам), а также схеме алгоритма. Это существенно упрощает запись и сокращает время составления и отладки программы. Однако взаимно-однозначное соответствие между блоками алгоритма и операторами (или операторами различных входных языков) не всегда соблюдается — в программе несколько операторов могут отображать один блок алгоритма и наоборот. Подобная ситуация характерна и для обычных языков, например, звуки, отображаемые в русском языке буквами «я» или «щ», описывают несколькими латинскими буквами, тогда как звук, описываемый в русском языке буквами «кс», отображают одной латинской буквой.

При составлении программ решения сложных задач часто предварительно составляют схему алгоритма, наглядно отображающую связь между блоками (пунктами описания или этапами вычислений), реализуемыми в программе одним или несколькими операторами. Функциональные операторные блоки в схемах алгоритмов обозначают прямоугольниками, связанными линиями со стрелками, указывающими последовательность выполнения операций. При этом переменные (операнды, значения которых могут изменяться в некоторых интервалах), а часто и постоянные величины обычно указывают не их численными значениями, а адресами регистров памяти, в которых хранятся эти значения. Это позволяет, в частности, изменять значения operandов, не изменения схемы алгоритма и соответствующей ему программы. Автоматическое прекращение вычислений с выводом результата последней операции в схемах алгоритмов обозначают овалами.

Рассмотрим вычисление функции y по формуле

$$y = \omega^2 LC / (1 + \omega^2 LC), \quad (134)$$

представляющей собой одну из возможных форм записи алгоритма вычисления y . Операции над числами ω , L и C по этой формуле при использовании микрокалькулятора с оператором вычисления обратной величины $1/x$ описываются на использовавшемся в предыдущей главе стандартизованном языке предложением

$$|C| \omega |x^2| \times |L| \times |C| + |1| |1/x| \times |\omega| \times |\omega| \times |C| \times |L| = .$$

Однако запись многозначных чисел непосредственно в программу нецелесообразна, так как ввод каждой цифры реализуется отдельным оператором и программа оказывается длинной. Поэтому исходные данные обычно предварительно заносят в регистры памяти, а в программу записывают операторы вызова содержимого этих регистров. Это позволяет изменять исходные данные, не изменяя программу, и сокращать ее длину.

Например, если занести числа ω , L и C соответственно в регистры 2, 3 и 4, то предложение для вычислений по формуле (134) примет вид

$$|C| |F2| |x^2| \times |F3| \times |F4| + |1| |1/x| \times |F2| \times |F2| \times |F3| \times |F4| = .$$

и отображает схему алгоритма, показанную на рис. 20, а.

Вычисления по формуле (134) можно реализовать без оператора $1/x$ согласно схеме алгоритма, показанной на рис. 20, б и описывающей предложением

$$|C| |F2| |x^2| \times |F3| \times |F4| + |1| |P5| |F2| |x^2| \times |F3| \times |F4| \div |F5| = .$$

Однако этот алгоритм избыточен, так как вычисление числа $\omega^2 LC$ повторяется дважды. Затраты времени и длину программы (или предложения в обычном режиме) можно уменьшить, более рационально используя регистры памяти (рис. 20, в):

$$|C| |F2| |x^2| \times |F3| \times |F4| |P5| + |1| |1/x| \times |F5| = .$$

Если число y определяют по формуле (134) для фиксированных значений операндов, целесообразно использовать обычный режим вычислений. Однако, если хотя бы один из операндов изменяется и необходимо найти значения y_i для ряда дискретных значений этого операнда в интервале его изменения, то следует использовать про-

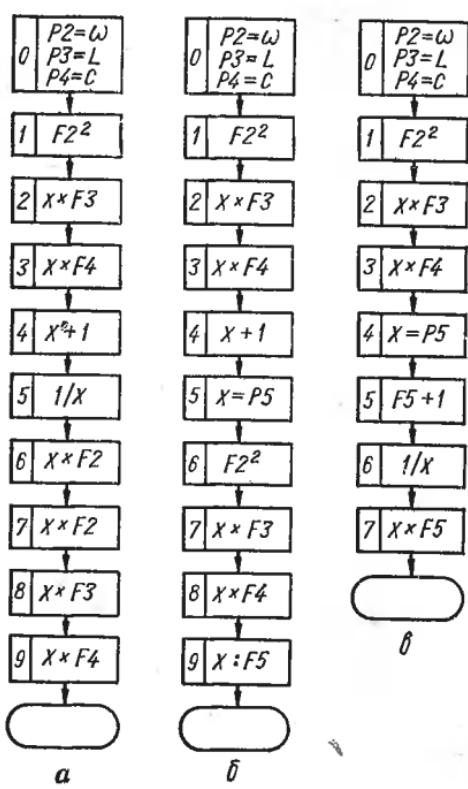


Рис. 20. Схемы алгоритмов вычислений по формуле (134)

граммрируемый режим с пуском программы для каждого нового значения изменяемого операнда.

В ЭВМ с печатающими выходными устройствами и достаточно большой емкостью запоминающих устройств результаты вычислений для ряда значений исходных operandов выводятся на печать или заносятся в память без прекращения вычислений. Однако в программируемых микрокалькуляторах вывод результата возможен, как правило, только на индикатор и вычисления приходится прерывать для регистрации в вычислительном бланке очередных результатов вычислений.

Дефиниции или определения, характерные для обычных языков (например «Это программируемый калькулятор» или «Мы — веселые ребята») в алгоритмических языках реализуются операторами присвоения вида $I = A$, где I — символ пе-

ременной (идентификатор), принимающей значение, равное результату вычисления выражения A в правой части. Оператор присвоения лишь по форме совпадает с обычной записью уравнения, и из двух идентичных в арифметике выражений вида $A + B = C$ и $C = A + B$ для машины имеет смысл только второе. Например, типичная для машинных языков запись вида $K = K + A$ означает, что переменной K в дальнейшем присваивается значение, на число A большее предыдущего.

В языках ряда микрокалькуляторов символ $=$ используется как символ присваивания результата предыдущих вычислений содержимому индикаторного регистра. Однако в некоторых входных языках этот символ отсутствует и операция присвоения реализуется другими операторами. Для этого обычно используют запись нового значения переменной в тот же регистр памяти, в котором хранилось прежнее значение. Например, формула присвоения

хранящейся в регистре K переменной нового значения $k = f(k)$ реализуется фрагментом программы вида $|FK| \dots |PK|$, где многоточие заменяет часть программы, вычисляющей новое значение переменной. В этом случае формула присвоения вида $K = K + A$ реализуется фрагментом программы $|FK| + |FA| |PK|$, а формула вида $K = K \times A$ — фрагментом $|FK| \times |FA| |PK|$.

В тех случаях, когда дискретные значения операндов в интервалах их изменения выбираются по некоторому закону, вычисления можно ускорить, автоматизируя процесс перехода к следующим значениям operandов с помощью оператора присвоения изменяемым operandам новых значений рассмотренным выше способом.

Для примера рассмотрим обычный случай разбиения интервала аргумента на n равных отрезков. Пусть, например, значение y вычисляется по формуле (134) при изменении числа ω от ω_n до ω_v в $n+1$ точках при фиксированных значениях переменных L и C . В этом случае можно предварительно вычислить шаг $\Delta\omega = (\omega_v - \omega_n) : n$ по заданному числу n , ввести значения $\omega = \omega_n$, L , C и $\Delta\omega$ соответственно в регистры 2, 3, 4, 5 и использовать в качестве программы (описывающей схему алгоритма, показанную на рис. 21, a) предложение вида

$$|C|F2|x^2|\times|F3|\times|F4|P6| + 1|P7|F2| + |F5|P2|F6|:|F7|=!.$$

При использовании такой программы после каждого вычисления y_i (и записи результата) содержимое регистра 2, равное ω_i , увеличивается на величину $\Delta\omega$ и, следовательно, для получения требуемых $n+1$ значений y_i достаточно $n+1$ раз выполнить программу при вводе исходных данных только перед началом вычислений.

Аналогичный алгоритм используется и при других законах изменения аргумента. Для часто встречающегося условия изменения операнда a по логарифмическому закону в $n+1$ точках очередное значение вычисляется по формуле

$$a_i = a_n (a_v/a_n)^{i/n} = a_n q^i; \quad i = 0, 1, 2, \dots, n, \quad (135)$$

где a_n и a_v — нижняя и верхняя границы интервала изменения a ; n число отрезков разбиения интервала.

Например, при вычислении $y_i = y(\omega_i)$ по формуле (134) при изменении ω_i по логарифмическому закону в $n+1$ точках целесообразно в соответствии с формулой (135) предварительно

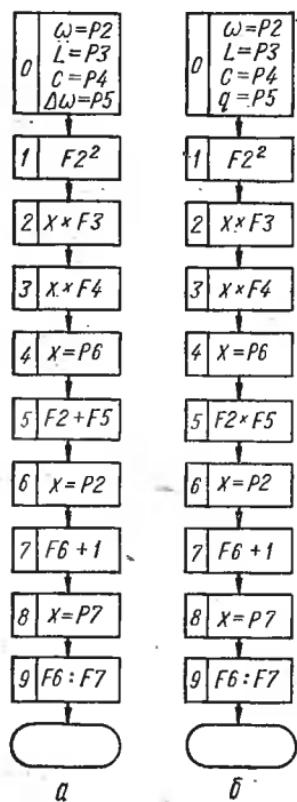


Рис. 21. Схема алгоритма вычисления функции при изменении аргумента

вычислить коэффициент q и использовать схему алгоритма, показанную на рис. 21, б.

Алгоритмические языки универсальных ЭВМ и программируемых микрокалькуляторов, кроме операторов ввода чисел и вычисления функций, содержат также операторы безусловных и условных переходов, позволяющие создавать разветвленные программы, значительно расширяющие возможности решения различного рода задач и ускоряющие вычисления.

Оператор безусловного перехода БП (в алгоритмических языках универсальных ЭВМ он обычно обозначается символами «ИДТИ НА» или «GO TO») передает управление любому другому оператору программы. Для этого после оператора перехода указывается адрес перехода, т. е. адрес в программе оператора (команды), которому передается управление. К безусловным относятся и операторы обращения к многократно используемой части программы (подпрограмме или процедуре) и выхода из нее.

Наиболее характерные операторы условного перехода или условные операторы отображают логические построения вида «если A , то B , иначе C », где A — некоторое условие для результата x предыдущей операции (например, $x > 0$, $x = 0$ и т. п.). Если условие A выполняется («Да»), то условный оператор передает управление по адресу B , в противном случае («Нет») — по адресу C . Использование условных операторов существенно расширяет возможности решения математических задач, обеспечивая, в частности, изменение хода вычислений в зависимости от промежуточных результатов, а также построение программным путем составных условных операторов с более сложными условиями или большим числом выходов.

Адреса переходов в программе указываются в различных алгоритмических языках некоторым символом («меткой»), стоящим перед оператором, которому передается управление, или кодами порядковых номеров шагов (команд), соответствующих операторам, которым передается управление.

В отличие от операторов безусловного перехода и обычных функциональных операторов или блоков операторы условного перехода обозначают в схемах алгоритмов ромбами с двумя выходами по адресам B («Да») и C («Нет»). Способ указания адресов переходов в программе зависит от грамматики входного языка ЭВМ или калькулятора используемого типа. В иллюстративных примерах настоящего раздела после оператора перехода в программе указан адрес безусловного перехода или условного перехода при невыполнении условия («Нет») в виде метки MN (N — номер метки), стоящей также перед оператором, которому передается управление. Если условие выполнено («Да»), то управление передается оператору, следующему в программе после адреса перехода «Нет». Оператор автоматического прекращения вычислений в этих иллюстративных программах обозначен символом $=$.

Типичным примером использования в программах условных операторов является формирование замкнутого цикла, вычисле-

ния в котором продолжаются при автоматическом изменении аргументов до выполнения заданного условия.

Пример 42. Составить программу вычислений корня уравнения $e^x - 10x = 0$ с точностью до $\epsilon = 0,1$.

Определим грубо интервал, в котором находится корень. При $x = 0$ невязка заданного уравнения равна единице, а при $x = 1$ отрицательна. Следовательно, корень находится в интервале $[0; 1]$ и его можно найти с требуемой точностью, вычисляя невязки уравнения для ряда значений $x_{i+1} = x_i + 0,05$; $i = 0, 1, 2, \dots$ при $x_0 = 0$ до тех пор, пока невязка станет отрицательной.

Выбранный способ решения при исходных данных $x_0 = 0$ и $\Delta x = 0,05$ описывается следующим алгоритмом:

1. Вычислить $x_{i+1} = x_i + \Delta x$.
2. Вычислить $e^{x_{i+1}}$.
3. Вычислить невязку $\delta = e^{x_{i+1}} - 10x_{i+1}$.
4. Если $\delta > 0$, то перейти к п. 1, иначе к п. 5.
5. Закончить вычисления, приняв в качестве искомого корня значение x_{i+1} , при котором нарушается условие $\delta > 0$.

Подобный алгоритм наглядно отображается схемой, показанной на рис. 22. Если вычисленная в блоке 6 невязка положительна, то в регистр 2 записывается новое значение $x_{i+1} = x_i + \Delta x$ и вычисления повторяются до нарушения условия, определяемого блоком 7, после чего вычисления автоматически прекращаются.

Таким образом, решение задачи обеспечивается программой вида

$$|C | M1 | F2 | + | F3 | P2 | \times | 10 | P4 | F2 | e^x | - | F4 | x < 0 | M1 | F2 | = |$$

с указанием адреса «Нет» условного перехода меткой M1 при исходных данных $x_0 = 0$ и $\Delta x = 0,05$, занесенных соответственно в регистры 2 и 3.

Для вычисления второго корня заданного уравнения достаточно в составленной программе заменить условный оператор на $x \geq 0$, а в качестве исходного принять значение $x_0 = 1$.

В одной и той же программе или даже в одном цикле можно использовать различные условные и безусловные операторы. С помощью подобных программ решают многообразные задачи, включая разбиение чисел по заданным классам, а также задачи, относимые обычно не к численным, а к логическим или структурным задачам. Для иллюстрации совместного использования различных операторов перехода рассмотрим простейшую задачу идентификации класса чисел.

Пример 43. Составить программу определения четности целого числа.

Результатом k -кратного вычитания числа 2 из четного числа $a = 2k$ будет нуль. Если же число $a = 2k + 1$ нечетное, то при любом числе вычитаний результат будет отличен от нуля, а при $(k + 1)$ -кратном вычитании будет отрицательным. Следовательно, можно составить следующий (хотя и не лучший в отношении затрат времени) алгоритм, при котором исследуемое число заносится в регистр 2 перед пуском программы:

1. Вычесть 2 из содержимого регистра 2.
2. Если результат равен нулю, перейти к п. 4, иначе к п. 3.
3. Если результат меньше нуля, перейти к п. 5, иначе к п. 1.

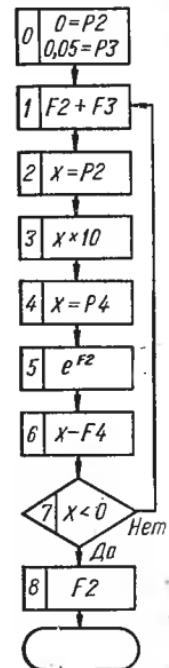


Рис. 22. Схема алгоритма решения уравнения

4. Вывести на выходное устройство число 0 (заданное число четное) и прекратить вычисления.

5. Вывести на выходное устройство число 1 (заданное число нечетное) и прекратить вычисления.

Составленный алгоритм наглядно отображается схемой, показанной на рис. 23. На используемом стандартизованном входном языке он описывается следующей программой:

| C | M1 | F2 | — | 2 | P2 | x = 0 | M2 | 0 | = M2 | x < 0 | M1 | 1 | = |

с введением заданного числа в регистр 2 перед пуском программы.

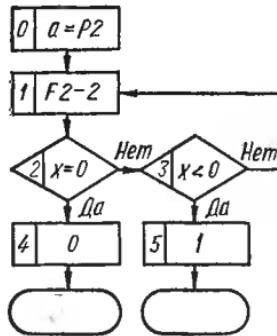


Рис. 23. Схема алгоритма определения четности числа

Основную часть схемы алгоритма до выводов «Да» включительно в рассмотренном примере можно рассматривать как схему составного оператора определения четности, передающего управление по различным адресам в зависимости от четности или нечетности заданного числа. Подобным образом с помощью условных операторов, содержащихся во входном языке, можно формировать и более сложные составные операторы.

Условные операторы облегчают решение задач выбора ситуации или структуры заданной системы, удовлетворяющей некоторому критерию качества (оптимальности). Подобные задачи в простейших случаях решают

методом полного перебора всех возможных вариантов решения с выбором наилучшего из них. Однако в большинстве случаев метод полного перебора непригоден в связи с большими затратами времени и емкости запоминающих устройств. Более эффективны эвристические («способствующие открытию») методы решения задач оптимизации структуры, в которых учитываются конкретные особенности исходной постановки задачи и ее возможных математических формулировок.

В развитии эвристического программирования большое значение имеют игровые задачи для двух партнеров (одним из которых является ЭВМ), решение которых заканчивается достижением выигрышной или ничейной ситуации. Составление программ сложных игр для микрокалькуляторов с ограниченной емкостью запоминающих устройств практически невозможно. Однако программирование простейших игр раскрывает многие потенциальные возможности микрокалькулятора используемого типа, которые можно успешно использовать при решении инженерных задач.

Для иллюстрации рассмотрим составление алгоритма для известной игры в «крестики — нолики». Она заключается в поочередном занятии партнерами клеток таблицы размера 3×3 до тех пор, пока одному из них удастся занять три клетки подряд по горизонтали, вертикали или диагонали или же это окажется невозможным для обоих. Несмотря на простоту этой игры, ее программирование даже для универсальных ЭВМ относительно сложно. При исполь-

зовании программируемых микрокалькуляторов трудности программирования усугубляются ограниченной емкостью запоминающих устройств, в связи с чем особое значение приобретает выбор наиболее эффективных алгоритмов.

Допустим, что игру начинает микрокалькулятор, занимая первым ходом центр таблицы с показанной на рис. 24, а нумерацией клеток. Выбор дальнейшей стратегии игры для микрокалькулятора зависит от ответного (первого) хода противника. Если этим ходом он занимает клетку с четным номером (в связи с симметрией таблицы относительно центра несущественно какую), то выигрыш микрокалькулятора обеспечен. Действительно, в этом случае вторым ходом микрокалькулятор занимает смежную занятой противником нечетную клетку, что вынуждает последнего следующим ходом занять клетку, закрывающую диагональ (рис. 24, б). Однако в этом случае микрокалькулятор третьим ходом занимает клетку, смежную ранее им занятой, что обеспечивает его выигрыш на следующем ходу. Если же противник первым ходом занимает нечетную клетку (рис. 24, в), то при отсутствии грубых ошибок игра неизбежно оканчивается вничью.

Пример 44. Составить алгоритм игры в «крестики — нолики», начинаящейся первым ходом микрокалькулятора в центр игровой таблицы (рис. 24, а).

Играющий, независимо от его первого хода, во избежание проигрыша, вынужден занимать определенные клетки. Поэтому для ответного хода в памяти микрокалькулятора достаточно хранить номер предыдущего хода (номер клетки) и ответного хода играющего.

Алгоритм игры начнем с вывода первого хода $y_1 = 9$ микрокалькулятора (блок 1 на рис. 25). Ответный ход x_1 играющего записывается блоком 2 в один из регистров памяти. Для занятия микрокалькулятором смежной клетки в блоке 3 организуется вычисление второго хода по формуле $y_2 = [x_1 - 1]_8$, означающей

$$y_2 = \begin{cases} x_1 - 1, & \text{если } x_1 - 1 > 0; \\ x_1 - 1 + 8, & \text{если } x_1 - 1 \leq 0. \end{cases}$$

Вычисленное значение y_2 заносится в память (блок 4) и выводится на индикатор (блок 5).

После второго хода играющего x_2 , записываемого вместо x_1 блоком 6, проверяется возможность выигрыша ходом на клетку, противоположную y_2 . Для этого в блоке 7 вычисляется значение $z = [y_2 - 4]_8$, которое сравнивается блоком 8 со значением x_2 . Если клетка z свободна ($z - x_2 \neq 0$), то игра заканчивается выигрышем микрокалькулятора $y_3 = z$. В противном случае оптимальная стратегия зависит от четности номера клетки x_1 , занятой играющим первым ходом (и, как следствие, последующих ходов). Проверка четности x_2 , противоположной четности x_1 , организуется блоком 10. Если x_2 четно,

1	2	3
8	9	4
7	6	5

а

0		
X		

X	0	
X	X	

X	0	
X	X	0

б

0		
X		

0		
0	X	X

X	0	
X	X	X

в

Рис. 24. Игровые таблицы и ситуации

то блок 11 вычисляет номер клетки $y_3 = [x_2 - 1]_8$, смежной y_2 против часовой стрелки. Если же x_2 нечетно, то блок 12 вычисляет выигрышный ход $y_3 = [y_2 - 1]_8$.

Дальнейший алгоритм игры можно либо разделить на две ветви в зависимости от предыдущей проверки четности, либо объединить обе стратегии с последующей проверкой четности номера клетки, занимаемой играющим. В последнем случае оба варианта игры объединяются блоками 13 и 14, запи-

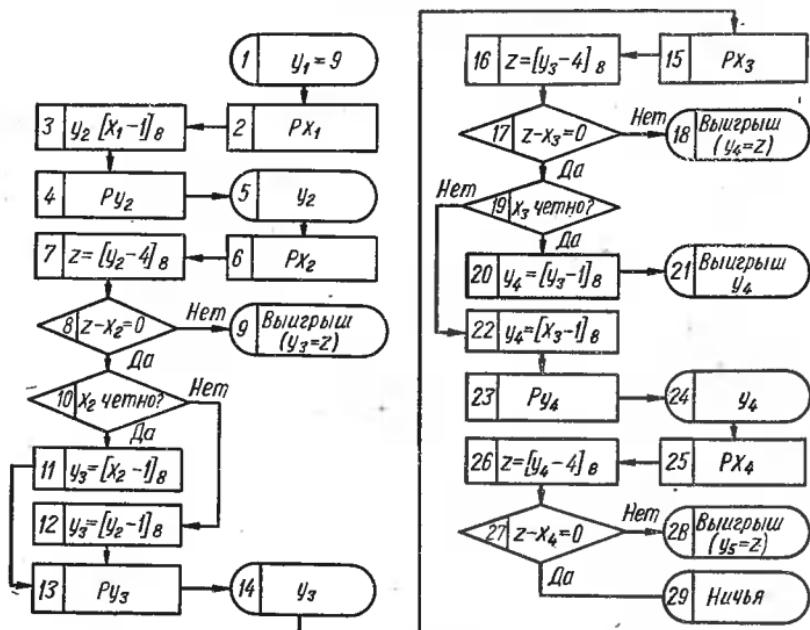


Рис. 25. Алгоритм игры

сывающими y_3 вместо y_2 и выводящими ответ микрокалькулятора на индикатор.

После записи третьего хода x_3 играющего (блоком 15) вместо x_2 проверяется возможность выигрыша ходом на клетку, симметричную y_3 (блоками 16, 17 и 18, идентичными соответственно блокам 7, 8 и 9). Если выигрыш не достигается, то блок 19 проверяет четность x_3 , совпадающую с четностью x_1 . Если x_3 четно, то форсированный выигрыш достигается занятием клетки с номером $y_4 = [y_3 - 1]_8$, вычисляемым и выводимым на индикатор блоками 20 и 21.

При нечетном x_3 блок 22 вычисляет номер клетки $y_4 = [x_3 - 1]_8$ (дающий при ошибке играющего возможность выигрыща), который записывается блоком 23 вместо y_3 и выводится блоком 24 на индикатор.

Четвертый ход играющего x_4 заносится блоком 25 в память вместо x_3 , после чего блоки 26, 27 и 28 проверяют возможность выигрыша ходом $y_5 = [y_4 - 4]_8$. Если выигрыш невозможен, то ход $y_5 = [y_4 - 1]$ обеспечивает ничейный результат игры.

Вычислительные блоки, используемые в рассмотренном примере, могут быть реализованы с помощью условных операторов типа $x = 0$, $x \neq 0$, $x \geq 0$ или $x < 0$, имеющихся во входных языках практически всех ЭВМ и программируемых микрокалькуляторов. Так, составной вычислительный блок для определения

$y_i = [x_i - A]_8$ при наличии условного оператора $x < 0$ реализуется следующим фрагментом программы на стандартном входном языке:

$$\dots | x_i | - | A | x < 0 | M1 | + | 8 | M1 | = | \dots .$$

Длину программы можно сократить при наличии ее в выходном языке микрокалькулятора используемого типа оператора обращения к подпрограмме, позволяющего совместить некоторые блоки приведенного в последнем примере алгоритма. При отсутствии подобного оператора его можно сформировать программным способом с помощью операторов условных и безусловных переходов и счетчика числа обращений, также реализуемого с помощью этих операторов.

Пусть, например, требуется трижды использовать одну и ту же часть программы (называемую в этом случае подпрограммой или процедурой), которую на схеме алгоритма обозначим символом ПП (блок 1 на рис. 26). Результат вычислений по этой подпрограмме записывается блоком 2 в регистр памяти 3. После первого обращения к подпрограмме счетчиком из блоков 3 и 4 в ранее свободный регистр 4 записывается число 1 и после вычисления $F4 - 2 = -1$ в блоке 5 условный оператор 6 передает управление последующей части программы с предварительным вызовом в блоке 7 результата вычислений по подпрограмме.

При втором обращении к подпрограмме результат предыдущих вычислений с помощью оператора безусловного перехода вводится в подпрограмму. В этом случае счетчик числа обращений заносит в регистр 4 число 2, а блок 5 вычисляет $F4 - 2 = 0$, вследствие чего операторы 6 и 8 передают управление блоку 9. Этот блок вызывает из регистра 3 результат вычислений по подпрограмме при втором обращении и вводит его в последующую часть программы.

При третьем обращении к подпрограмме оператор безусловного перехода вводит результат предыдущих вычислений в подпрограмму, в блоке 5 вычисляется $F4 - 2 = 1$ и управление передается блоку 10, который вводит результат вычислений по подпрограмме при третьем обращении в следующую часть программы.

Схема описанного алгоритма троекратного обращения на стандартном входном языке реализуется фрагментами программы вида $\dots | P2 | M1 | \text{ПП} | P3 | F4 | + | 1 | P4 | - | 2 | x < 0 | M2 | F3 | \dots | P2 | \text{БП} | M1 | M2 | x = 0 | M3 | P3 | \dots | P2 | \text{БП} | M1 | M3 | P4 | \dots ,$

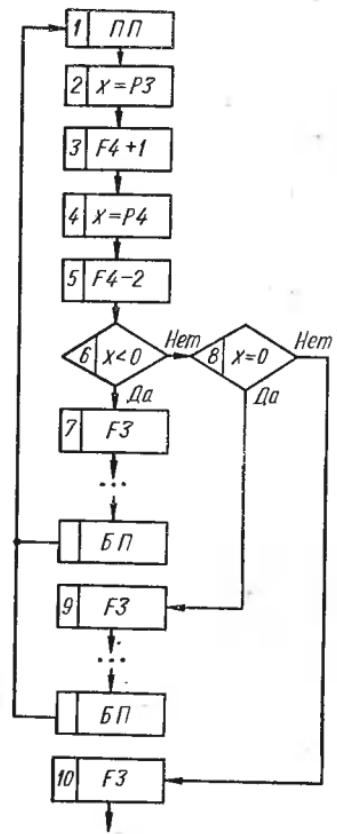


Рис. 26. Схема алгоритма троекратного обращения к подпрограмме

где М1, М2 и М3 — метки операторов, а ГП — обобщенный символ подпрограммы.

Подобным образом можно реализовать вычисления по подпрограмме и при другом числе обращений. Однако алгоритмы обращения к подпрограмме целесообразно использовать лишь в тех случаях, когда их применение сокращает общую длину программы или время счета.

Длину программы часто удается сократить при наличии во входном языке микрокалькулятора операторов, облегчающих выполнение операций над множествами (последовательностями или массивами) чисел, хранимых в запоминающих устройствах. Некоторые из таких операторов позволяют выводить на индикатор и выводить из индикаторного регистра последовательности хранимых чисел. Еще более эффективны операторы, позволяющие выполнять операции над номерами (кодами) ячеек или регистров памяти, в которых хранятся операнды.

Объем запоминающих устройств программируемых микрокалькуляторов ограничен. Кроме того, хотя зависимость между числом шагов программы и временем счета определяется видом операторов программы, в общем случае при сокращении длины программы уменьшается и время счета. Поэтому при решении относительно сложных задач необходимо использовать все возможности для составления короткой программы, требующей минимальных затрат времени на вычисления.

Следует учитывать, что не существует твердых правил составления алгоритмов для решения задач различных классов, поэтому программирование остается в значительной мере искусством. В связи с этим для составления удовлетворительной программы решения сравнительно сложной задачи приходится ватрачивать много времени на выбор надлежащей формулировки задачи, формирование алгоритма ее решения, программную реализацию алгоритма и редактирование и отладку программы. Однако эти усилия, как правило, окупаются при использовании составленной программы для многократных вычислений.

2. ОСОБЕННОСТИ ПРОГРАММИРУЕМЫХ МИКРОКАЛЬКУЛЯТОРОВ

С позиций пользователя программируемые микрокалькуляторы отличаются от универсальных ЭВМ главным образом лексикой и грамматикой входных языков, количественными показателями — массой и габаритными размерами, быстродействием, емкостью и количеством запоминающих устройств различного типа, а также числом и типами устройств ввода и вывода информации.

Быстродействие калькулятора зависит как от физических факторов, определяющих механизм работы его интегральных структур, так и от выбранного конструкторами способа обмена информацией между его различными устройствами. Современные электронные ми-

рекалькуляторы в отношении быстродействия существенно уступают ЭВМ высокой производительности. В большинстве калькуляторов выполнение арифметической операции обычно длится десятые доли секунды, что на несколько порядков больше затрат времени на выполнение той же операции на универсальной ЭВМ. Поэтому при решении сложных задач на микрокалькуляторе особенно существенно составление экономных программ автоматических вычислений, реализуемых при минимальных затратах времени.

Максимальное время автоматических вычислений по введенной в калькулятор программе ориентировочно оценивают, суммируя затраты времени на выполнение отдельных ее операций. Для этого используют данные о скорости выполнения различных операций, приводимые в инструкции по эксплуатации микрокалькулятора используемого типа. Точная оценка подобного рода не всегда возможна, так как при наличии в программе замкнутых циклов заранее неизвестно число операций, производимых микрокалькулятором для получения результата. Однако даже грубая оценка предполагаемого времени вычислений может избавить пользователя от напрасного ожидания результата при зацикливании плохо отлаженной программы. Поэтому составляемые программы необходимо дополнять указанием на время ее выполнения для контрольных (тестовых) примеров.

Программируемые микрокалькуляторы могут иметь устройства памяти для хранения operandов и результатов операций как в виде запоминающего устройства с произвольной выборкой (ЗУПВ), позволяющего независимо использовать каждый его регистр, так и в виде стеков, вызов содержимого регистров которых возможен лишь в определенной последовательности. Суммарное число регистров памяти (ЗУПВ и стеков) в программируемых микрокалькуляторах массовых типов обычно не превышает 10—20 и лишь в специализированных микро-ЭВМ этого типа превосходит 100.

Характерной особенностью программируемых калькуляторов является наличие устройства запоминания программы (УЗП) или программной памяти, куда заносится программа автоматических вычислений. Емкость программной памяти обычно оценивают максимальным числом шагов (команд) запоминаемой программы, определяющим и предельные размеры задач, решаемых в автоматическом режиме. В большинстве программируемых микрокалькуляторов емкость УЗП ограничена 50—250 шагами. Рациональные пределы емкости запоминающих устройств калькулятора непосредственно связаны со способом ввода программы в УЗП. В различных типах программируемых микрокалькуляторов программа вводится отдельными командами нажатием соответствующих клавишей и автоматически стирается при вводе новой программы или при выключении микрокалькулятора. Поэтому относительно длительную и требующую большого внимания процедуру ввода программы нажатием клавишей приходится повторять перед каждым ее использованием. Это обстоятельство ограничивает сверху целесообразный предел

емкости УЗП и, следовательно, объем задач, решаемых в автоматическом режиме.

Недостатки ручного ввода программ устраниены в некоторых микрокалькуляторах и микро-ЭВМ настольных типов, предусматривающих возможность автоматического ввода программы от внешних накопительных устройств (содержащих, например, кассету с магнитной лентой). Однако подобный способ, характерный для универсальных ЭВМ, ограничивает автономность и портативность вычислительного устройства, вследствие чего непригоден для микрокалькуляторов «карманного» типа.

В некоторых наиболее совершенных микрокалькуляторах проблема постоянного хранения программы и занесения ее в УЗП решается путем записи программы на малогабаритных носителях (обычно пластинка с магнитным покрытием), вводимых в микрокалькулятор для смены программы. Подобное техническое решение позволяет значительно увеличить емкость запоминающих устройств и по степени автоматизации вспомогательных операций приближает микрокалькулятор к ЭВМ больших габаритов. К сожалению, подобная модификация калькулятора отличается весьма высокой стоимостью, что в настоящее время существенно ограничивает ее практическое применение.

Ввод исходных данных во всех микрокалькуляторах реализуется нажатием клавиш с визуальным контролем правильности ввода по индикатору. Как исключение, в некоторых типах специализированных микрокалькуляторов предусмотрены дополнительные электрические разъемы или другие устройства для ввода информации без помощи клавиш.

Основным и для большинства микрокалькуляторов единственным устройством вывода данных является цифровой индикатор на светодиодах или жидкких кристаллах. При этом в программе можно предусматривать занесение результатов вычислений в регистры памяти, но для считывания они должны быть вызваны в индикаторный регистр. Исключение составляют некоторые типы микрокалькуляторов, в которых имеются разъемы для подключения внешних печатающих устройств или даже встроенное миниатюрное печатающее устройство. Однако подобные устройства в основном характерны для микро-ЭВМ настольных типов, а не для микрокалькуляторов.

Программируемые микрокалькуляторы различных типов отличаются лексикой и грамматикой входного языка. Основные грамматические различия входных языков связаны со способами указания в программе связи между operandами и операторами, указания адресов условных и безусловных переходов, а также способами обмена содержимым запоминающих устройств.

Одноместные операторы в большинстве входных языков помещаются в программе после операнда, описанного численным значением содержимого индикаторного регистра (которое может быть результатом предыдущей операции или операции вызова содержимого другого регистра памяти). Для двухместных операций харак-

терны два основных способа связи операторов с операндами. Первый из них аналогичен обычной математической записи арифметических операций вида yAx , где символ A оператора записывается между операндами, причем первый из них может быть результатом предыдущей операции и в программе не записывается. В этом случае соответствующая оператору A операция выполняется после ввода оператора, следующего в программе после второго из операндов (таким оператором, в частности, может быть оператор $=$, вызывающий результат операции в индикаторный регистр).

При втором способе символ двухместного оператора A записывают в программе после операндов, разделенных оператором \uparrow , за-сылающим первый из операндов из индикаторного в рабочий (операционный) регистр (например $y \uparrow xA$, где y и x — операнды, над которыми выполняется двухместная операция). В этом случае операция выполняется непосредственно после ввода второго операнда в индикаторный регистр. Использование этого способа связи операндов и оператора несколько повышает быстродействие микрокалькулятора в программируемом режиме, но для отображения в программе двухместной операции требуется на одну команду больше, чем при первом способе. Для сокращения длины программы во входных языках некоторых микрокалькуляторов со вторым способом описания двухместных операций допускается исключение из этого правила — символ оператора \uparrow можно опускать перед операндом, записанным непосредственно в программу.

Грамматические правила, связанные со способом указания адресов условных и безусловных переходов в программе, а также со способами обмена содержимым регистров памяти, для входных языков программируемых микрокалькуляторов различных типов более разнообразны, так как определяются конструктивными особенностями запоминающих устройств и устройств обмена информацией. Особенно характерно для входных языков программируемых микрокалькуляторов наличие префиксных операторов, обеспечивающих использование одних и тех же клавиш для ввода различных функциональных операторов.

Входным языком высокого уровня с относительно развитой лексикой отличается распространенный программируемый микрокалькулятор типа «Электроника Б3-21». Словарный запас входного языка этого микрокалькулятора определяется символами операторов, указанных на клавиатуре (рис. 27). Для сокращения числа клавиш все операторы разбиты на три группы. Операторы первой группы, символы которых указаны непосредственно на клавишиах, вводят нажатием соответствующих клавиш. Для ввода операторов второй группы, символы которых указаны под клавишами нижнего ряда и над клавишами остальных рядов, необходимо предварительно нажать префиксную клавишу Р. Операторы третьей группы, символы которых указаны над клавишами нижнего ряда, вводят после предварительного нажатия клавиши F. Кроме того, для засылки содержимого индикаторного регистра в регистры памяти и вызова их

содержимого, как описано ниже, также используются префиксные клавиши Р и F.

В микрокалькуляторе рассматриваемого типа, кроме индикаторного регистра X, имеется кольцевой стек с шестью регистрами и запоминающее устройство с произвольной выборкой (ЗУПВ), содержащее восемь регистров памяти. Первый из регистров ЗУПВ (регистр 1 или Y) используется в качестве операционного (рабочего) регистра, куда заносится первый операнд при двухместных операциях. Поэтому для постоянного хранения исходных данных и результатов промежуточных вычислений можно использовать 13 регистров — шесть регистров стека и семь регистров ЗУПВ с номерами 2—8.

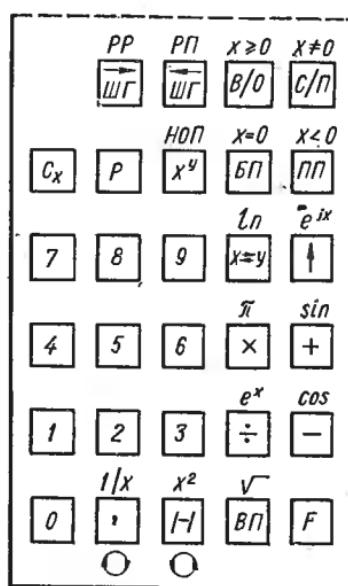


Рис. 27. Клавиатура программируемого микрокалькулятора «Электроника Б3-21»

Работа с микрокалькулятором типа «Электроника Б3-21» в обычном и программируемом режиме характеризуется следующими основными особенностями:

1. В микрокалькулятор можно вводить числа с фиксированной или плавающей запятой. Представление чисел с плавающей запятой ограничено пределами от $A_{\min} = \pm 1 \cdot 10^{-99}$ до $A_{\max} = \pm 9,9999999 \cdot 10^{99}$. Вычисления производятся над десятичными представлениями чисел с 8-разрядной мантиссой, но для дробных и смешанных чисел на индикаторе высвечиваются только семь значащих цифр (включая 0 для целой части дробных чисел). Для высвечивания на индикаторе цифры восьмого разряда в этом случае достаточно вычесть $a_m \cdot 10^n$, где a_m — первая значащая цифра (например, при вызове числа π на индикаторе высвечивается 3,141592, но при вычитании из него числа 3 получим $1,1415927 \cdot 10^{-1}$).

2. Для ввода числа в индикаторный регистр последовательно нажимают клавиши, соответствующие операторам ввода цифр и, при необходимости, запятой, порядка числа ВП и изменения знака /—/ (при отрицательном знаке мантиссы или порядка числа этот оператор вводят соответственно после ввода цифр мантиссы или

порядка). Отрицательный знак высвечивается в крайнем левом, а цифры порядка и его знак — в крайних правых разрядах индикатора.

Содержимое индикаторного регистра стирается вводом оператора C_x , после чего вводят новое число. Для исправления порядка числа достаточно ввести правильное значение, так как при этом предыдущее значение поразрядно смещается влево и стирается.

3. Содержимое индикаторного регистра перезаписывается в регистры ЗУПВ операторами вида PN , где N — номер регистра ЗУПВ. Вызов числа из регистра N в регистр X реализуется оператором FN . Кроме того, число, занесенное в индикаторный регистр, заносится в регистр Y (регистр 1) оператором переноса \uparrow . Во всех этих случаях содержимое исходного регистра сохраняется до ввода нового числа.

При вводе числового коэффициента (не вызываемого из регистров памяти) после выполнения любой операции ее результат автоматически переносится из регистра X в регистр Y как в обычном, так и в программируемом режимах. Например, при выполнении фрагмента программы вида ... $FN\ 2 + \dots$ после вызова из регистра N первого операнда команда ввода числа 2 обеспечивает автоматический перенос (без участия оператора \uparrow) первого операнда в регистр Y , и выполнение оператора $+$ приводит к образованию в регистре X суммы $FN + 2$, тогда как в регистре Y сохранится значение FN .

Оператор $X \rightleftharpoons{} Y$ обменивает содержимое регистров X и Y , что удобно использовать для проверки содержимого регистра Y , а также в таких случаях, когда вызываемую переменную FN нужно, например, разделить (... $\uparrow\ FN\ X \rightleftharpoons{} Y \div \dots$) на результат предыдущей операции.

Обмен содержимым регистров X и кольцевого стека реализуется операторами поворота стека \rightarrow (по часовой стрелке) и \leftarrow (против часовой стрелки*). В первом случае содержимое регистра X переносится в регистр $C1$ стека (рис. 28), содержимое остальных регистров стека смещается по часовой стрелке и в регистр X переносится содержимое регистра $C6$. Во втором случае содержимое регистра X переносится в регистр $C6$, а в регистр X переносится содержимое регистра $C1$, причем содержимое остальных регистров стека соответственно смещается против часовой стрелки. Регистры памяти очищаются занесением нуля из индикаторного регистра. При выключении микрокалькулятора содержимое всех запоминающих устройств (включая УЗП) стирается.

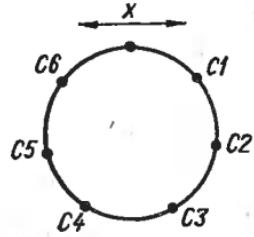


Рис. 28. Схема кольцевого стека

* В настоящей книге используется упрощенный вариант символов операторов поворота стека, отличающийся от указанного на клавиатуре. Кроме того, в программах символ $X \rightleftharpoons{} Y$ заменен упрощенным знаком XY .

4. Одноместные операторы \ln , e^{ix} , e^x , \sin , \cos (требующие предварительного нажатия клавиши Р) и x^2 , $1/x$, \sqrt{x} (требующие предварительного нажатия клавиши F) вводят после операнда, которым служит содержимое регистра X, в частности, результат предыдущей операции или вызванная оператором π постоянная 3,1415927.

Аргумент тригонометрических функций должен быть выражен в радианах, в связи с чем выраженный в градусах аргумент α следует предварительно преобразовать предложением вида $\alpha \uparrow \pi \times x \times 180 \div$. Результат одноместной операции e^{ix} заносится в регистры X и Y соответственно в виде значений $\cos x$ и $\sin x$. Поэтому для вычисления функции $\operatorname{tg} x$ достаточно дополнительно ввести оператор деления \div , а для вычисления функции $\operatorname{ctg} x$ — операторы $X \Leftrightarrow Y$ и \div .

Двухместные операторы $+$, $-$, \times , \div и X^y вводят после ввода обоих operandов, разделенных оператором \uparrow (исключение составляет рассмотренный выше случай числового значения второго операнда, когда оператор \uparrow можно не вводить). При вычислении функции X^y первым (как следует из символа оператора) вводят operand y .

5. Составляемая программа автоматических вычислений, кроме рассмотренных операторов вычисления функций, ввода чисел и переноса содержимого регистров памяти, может содержать операторы безусловного перехода БП и ПП (обращение к подпрограмме), а также операторы условного перехода $x = 0$, $x \neq 0$, $x < 0$ и $x \geqslant 0$. После каждого оператора перехода в программу записывают оператор передачи управления по адресу безусловного перехода или условного перехода «Нет» (условие не выполняется). Операторы передачи управления отображают символами одной или двух клавиш (в последнем случае первым будет символ Р или F префиксной клавиши), которые нужно нажать для перехода на требуемый шаг программы, имеющий адрес $a_i b_i$.

Так как первая команда программы записывается в ячейку УЗП с номером 00 = ab , то целесообразно принять адресацию шагов (операторов или команд) программы, несколько отличной от рекомендуемой инструкцией по эксплуатации микрокалькулятора рассматриваемого типа. В дальнейшем адресом i -го шага программы будем называть число $a_i b_i = a(b + 1)$, где $a_i = 0, 1, \dots, 9$ и $b = 1, 2, \dots, 6$. В этом случае в ячейки УЗП с номерами $a0$ или $a5$ будет заноситься код оператора, записанного по адресу шага программы $a1$ или ab соответственно. Например, в ячейках УЗП с номерами 05 и 70 будут храниться коды операторов, записанных в программе по адресам шагов 06 и 71 соответственно. При полном использовании емкости УЗП операторы программы имеют адреса от 01 до 96. Порядковые номера шагов программы, их адреса и символы операторов передачи управления по этим адресам приведены в табл. 39.

После оператора передачи управления (следующего за оператором условного или безусловного перехода) записывают следующую часть

Таблица 39

Адрес шага	Код оператора	Клавиша	Оператор	Операция
01		P 0		—
02		F 0		—
03	03	P ↑	e ^{ix}	Вычисление функции e ^{ix}
04	04	0	0	Ввод цифры 0 в регистр X
05		F ↑		—
06(10)	06	↑	↑	Засылка числа x в регистр 1 (Y)
11	11	P 1	P1	Засылка числа x в регистр 1 (Y)
12	12	F 1	F1	Вызов содержимого регистра 1 (Y)
13	13	P X↔Y	In	Вычисление функции In x
14	14	1	1	Ввод цифры 1 в регистр X
15		F X↔Y		—
16(20)	16	X↔Y	X↔Y	Обмен содержимым регистров X и Y
21	21	P 2	P2	Занесение числа x в регистр 2
22	22	F 2	F2	Вызов содержимого регистра 2
23	23	P ×	π	Вызов числа π в регистр X
24	24	2	2	Ввод цифры 2 в регистр X
25		F ×		—
26(30)	26	×	×	Умножение
31	31	P 3	P3	Занесение числа x в регистр 3
32	32	F 3	F3	Вызов содержимого регистра 3
33	33	P ÷	e ^{ix}	Вычисление функции e ^{ix}
34	34	3	3	Ввод цифры 3 в регистр X
35		F ÷		—
36(40)	36	÷	÷	Деление
	38	X ^y	X ^y	Вычисление функции xy
	39	P X ^y	НОП	Очистка шага программы
41	41	P 4	P4	Занесение числа x в регистр 4
42	42	F 4	F4	Вызов содержимого регистра 4
43	43	P ,	→	Поворот стека по часовой стрелке
44	44	4	4	Ввод цифры 4 в регистр X
45	45	F ,	1/x	Вычисление обратной величины
46(50)	46	,	,	Ввод запятой в регистр X
	48	B/O	B/O	Возврат из подпрограммы
	49	P B/O	x ≥ 0	Проверка условия x ≥ 0
51	51	P 5	P5	Занесение числа x в регистр 5
52	52	F 5	F5	Вызов содержимого регистра 5
53	53	P (—)	←	Поворот стека против часовой стрелки
54	54	5	5	Ввод цифры 5 в регистр X
55	55	F (—)	x ²	Возведение в квадрат
56(60)	56	(—)	(—)	Изменение знака
	58	БП	БП	Безусловный переход
	59	P БП	x = 0	Проверка условия x = 0
61	61	P 6	P6	Занесение числа x в регистр 6
62	62	F 6	F6	Вызов содержимого регистра 6
63		P ВП		—
64	64	6	6	Ввод цифры 6 в регистр X
65	65	F. ВП	V	Извлечение квадратного корня
66(70)	66	ВП	ВП	Ввод порядка числа x
	68	ПП	ПП	Обращение к подпрограмме
	69	P ПП	x < 0	Проверка условия x < 0
71	71	P 7	P7	Занесение числа x в регистр 7
72	72	F 7	F7	Вызов содержимого регистра 7

Адрес шага	Код оператора	Клавиша	Оператор	Операция
73		P C _x		—
74	74	7	7	Ввод цифры 7 в регистр X
75		F C _x		Очистка регистра X
76(80)	76	C _x	C _x	Остановка и пуск программы
	78	C/П	C/П	Проверка условия $x \neq 0$
	79	P C/П	$x \neq 0$	Занесение числа x в регистр 8
81	81	P 8	P8	Вызов содержимого регистра 8
82	82	F 8	F8	Вычисление функции cos x
83	83	P —	cos	Ввод цифры 8 в регистр X
84	84	8	8	—
85	85	F —		Вычитание
86(90)	86	—	—	—
91		P 9		—
92		F 9		—
93	93	P +	sin	Вычисление функции sin x
94	94	9	9	Ввод цифры 9 в регистр X
95		F +		—
96(—0)	96	+ ШГ ШГ Р РП Р РР	+ ШГ ШГ РП РР	Сложение Смещение программы на шаг вперед Смещение программы на шаг назад Переход в режим программирования Переход в рабочий режим

программы, причем при использовании условных операторов после оператора передачи управления по адресу «Нет» эта часть программы начинается с оператора, которому передается управление по адресу «Да». В конце программы записывается оператор С/П (стоп/пуск).

Подпрограммы обычно записываются после оператора С/П и заканчиваются операторами выхода из подпрограммы В/О (возврат/очистка), которые передают управление оператору, следующему в программе после оператора передачи управления по адресу подпрограммы. С помощью оператора ПП и следующего за ним оператора передачи управления возможно обращение как к первому, так и любому другому оператору подпрограммы.

6. Для ввода в УЗП составленной программы нажимается клавиша В/О, если программа заносится с первой ячейки, или клавиша БП и клавиши, кодирующие адрес шага, с которого заносится программа. Затем нажимаются клавиши Р и РП, что переводит микрокалькулятор в режим программирования, и в счетчике шагов (двух крайних правых разрядах, предназначенных для высвечивания порядка числа в рабочем режиме) высвечивается номер ячейки, с которой будет заноситься программа.

Программа заносится в УЗП последовательным введением операторов посредством нажатия тех же клавиш, что и в обычном ре-

жиме вычислений. При этом на индикаторе высвечиваются двухзначные коды трех введенных последними операторов, а в счетчике шагов — адрес оператора, введенного последним. Следует учитывать, что при введении оператора с адресом вида ab на счетчике высвечивается число $(a + 1)0$ (например, при введении оператора с адресом 26 в счетчике высвечивается число 30). Такие числа указаны в табл. 39 в круглых скобках. В этой же таблице приведены коды операторов, надписи на нажимаемых для их ввода клавишах, символы операторов и краткое описание их назначения (семантики). Четыре оператора кодов не имеют и используются только в режиме программирования или для перехода в другой режим.

При переполнении УЗП, когда число операторов программы превышает 60, в счетчике шагов высвечиваются отрицательные числа, а вводимые операторы записываются в начальную часть хранимой программы вместо занесенных ранее.

7. Для проверки и редактирования программы используются операторы $\overleftarrow{ШГ}$ и $\overleftarrow{ШГ}$; каждый ввод которых сдвигает коды операторов программы на индикаторе на один шаг вперед или назад. Если искомый оператор расположен в программе далеко от высвечиваемых, то следует нажать клавиши Р, РР, БП и клавиши, кодирующие номер шага, на единицу большего адреса нужного оператора. В этом случае при переходе в режим программирования (клавиши Р и РР) нужный оператор отображается своим кодом в крайних левых разрядах индикатора.

Для исправления замеченной ошибки в коде оператора, высвечиваемого в крайних левых разрядах, достаточно нажатием клавиши $\overleftarrow{ШГ}$ сдвинуть программу на шаг назад и ввести нужный оператор вместо ошибочного. Если занесенный ошибочно в программную память оператор нужно исключить, не затрагивая последующую часть программы, то вместо него вводится оператор НОП («нет операции»).

8. Для отладки занесенной в УЗП программы микрокалькулятор переводится операторами Р и РР в рабочий режим и в соответствующие регистры памяти заносятся исходные данные. После нажатия клавиши В/О (если программа должна выполняться с первого шага) или клавиши БП и клавишей, кодирующей переход на нужный шаг, последовательно нажимается клавиша ГП и по индикатору проверяется выполнение каждого шага программы.

9. Вычисления по занесенной в УЗП программе выполняются после перехода в рабочий режим (клавиши Р и РР), ввода в регистры памяти исходных данных, нажатия клавиши В/О (если программа выполняется с первого шага) или клавиши БП и клавиш, кодирующих адрес шага, с которого должно начинаться выполнение программы, и клавиши С/П.

При нажатии в рабочем режиме только клавиши С/П программа выполняется с шага, на котором была остановлена программа или закончены ее проверка или введение. В этом случае просматрива-

ются все последующие ячейки памяти (что увеличивает время выполнения программы или приводит к выполнению программы не с ее начала) и только после этого программа выполняется с первого шага, если в последующих ячейках программы не хранится оператор С/П.

Выполнение программы сопровождается мерцанием индикатора. В случае зацикливания программы, определяемого отсутствием результата в ожидаемый срок, вычисление следует прекратить нажатием клавиши С/П.

Превышение результатом операции верхнего предела представления чисел A_{\max} , перенос подобного числа из регистра X в регистр Y и попытки выполнить некорректную операцию (например, $\ln x$, $\sqrt{-x}$ или x^y при $x < 0$) приводят к переполнению и в крайних левых разрядах индикатора высвечивается —0 или число, начинающееся с одного или двух нулей. При попытке выполнить некорректную операцию операнды сохраняются в регистрах памяти и после исправления ошибки вычисления можно продолжить.

Составителю программы необходимо также учитывать, что при вычислении функций $\sin x$, $\cos x$, e^x , e^{ix} и X^y в разряды знака порядка некоторых регистров стека могут заноситься произвольные коды, что не позволяет использовать эти регистры при вычислениях указанных функций. Однако момент занесения зависит от времени, затрачиваемого на приведение аргумента в область сходимости ряда, по которому автоматически вычисляется функция. Это обстоятельство допускает определение регистров стека, которые не могут использоваться, в зависимости от значений аргумента функции. Поэтому при известном интервале аргумента функции использование стека допустимо за исключением регистров, номера которых и интервалы допустимых значений аргумента приведены в табл. 40. В связи со сложностью определения таких значений аргументов функции X^y они не приведены в этой таблице и при вычислении такой функции стек использовать не следует (см. приложение).

Таблица 40

$\sin x$, $\cos x$		e^x		e^{ix}	
Интервал x	$CN_{\text{пер}}$	Интервал x	$CN_{\text{пер}}$	Интервал x	$CN_{\text{пер}}$
$[0, \pi]$	6	$[0, 3 \ln 10]$	—	$[0, \pi]$	—
$[\pi, 3\pi]$	2	$[3 \ln 10, 9 \ln 10]$	2	$[\pi, 3\pi]$	4
$[3\pi, 10]$	—	$[9 \ln 10, 10 \ln 10]$	—	$[3\pi, 10]$	—
$[10, 5\pi]$	3	$[10 \ln 10, 12 \ln 10]$	6	$[10, 5\pi]$	5
$[5\pi, 7\pi]$	—	$[12 \ln 10, 12 \ln 10]$	1	$[5\pi, 7\pi]$	1
$[7\pi, 9\pi]$	5	$[12 \ln 10, 13 \ln 10]$	6	$[7\pi, 9\pi]$	—
$[9\pi, 10\pi]$	1	$[13 \ln 10, 14 \ln 10]$	1	$[9\pi, 10\pi]$	3
$[10\pi, 11\pi]$	4	$[14 \ln 10, 15 \ln 10]$	6	$[10\pi, 11\pi]$	6
$[11\pi, 13\pi]$	—	$[15 \ln 10, 16 \ln 10]$	1	$[11\pi, 13\pi]$	2
$[13\pi, 16\pi]$	6	$[16 \ln 10, 17 \ln 10]$	6	$[13\pi, 15\pi]$	—

В первых образцах микрокалькулятора рассматриваемого типа неконтролируемое переполнение возникает при сложении или вычитании чисел, из которых меньшее по модулю содержит семь девятерок и цифру, большую 4 восьмом разряде. Например, при сложении чисел 9,9999996 (на индикаторе высвечивается 9,999999) и 50 получается 160 вместо правильных 60. В последующих образцах этот недостаток устранен.

При использовании оператора обращений к подпрограмме ПП следует учитывать, что в случае его записи в ячейки ЗУП с номерами 55, 65, 70, 80, 81, 91 и 92 (что, согласно принятой нами адресации, соответствует шагам программы с адресами 56, 66, 71, 81, 82, 92 и 93) этот оператор выполнятся не будет.

Программируемый микрокалькулятор типа «Электроника Б3-34» отличается от рассмотренного более обширным словарным запасом (рис. 29), большей емкостью программной памяти и рядом особенностей, повышающих эффективность его использования. Этот микрокалькулятор автоматически выполняет вычисления по программам, содержащим до 98 операторов, имеет 14 регистров ЗУПВ и одновходовый стек, содержащий операционные регистры X и Y и вспомогательные регистры Z и T. Кроме того, имеется регистр X1 восстановления предыдущего результата, в который автоматически переписывается содержимое регистра X перед выполнением любой операции.

В адресной системе микрокалькулятора рассматриваемого типа использованы двухзначные десятичные числа (вместо кодов операторов у микрокалькулятора типа «Электроника Б3-21»), что упрощает составление программ и обеспечивает возможность изменения адресации переходов и обращений к ЗУПВ в процессе автоматических вычислений.

Словарный запас микрокалькулятора типа «Электроника Б3-34», кроме операторов входного языка микрокалькулятора типа «Электроника Б3-21» (за исключением операторов поворота стека и вычисления e^{ix}), включает операторы вычисления функций 10^x , $\lg x$, $\arcsin x$, $\arccos x$, $\arctg x$; операторы-счетчики L0, L1, L2, L3, а также операторы косвенных переходов — безусловного КБПН,

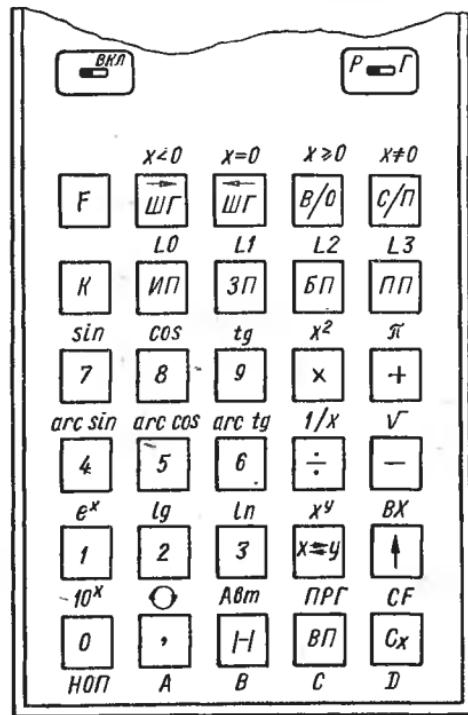


Рис. 29. Клавиатура программируемого микрокалькулятора «Электроника Б3-34»

условных $Kx < 0N$, $Kx \geq 0N$, $Kx = 0N$, $Kx \neq 0N$ и обращения к подпрограмме КППН (где К — префиксная клавиша, N — номер регистра ЗУПВ). Кроме того, имеется оператор BX восстановления результата предыдущей операции.

При выполнении операторов-счетчиков из содержимого регистров 0, 1, 2 и 3 соответственно вычитается единица и полученный результат проверяется по выполнению условия $x = 0$. Если это условие не выполняется, то управление передается по адресу, записанному в программе после оператора-счетчика. В противном случае продолжается выполнение программы, записанной после адреса перехода.

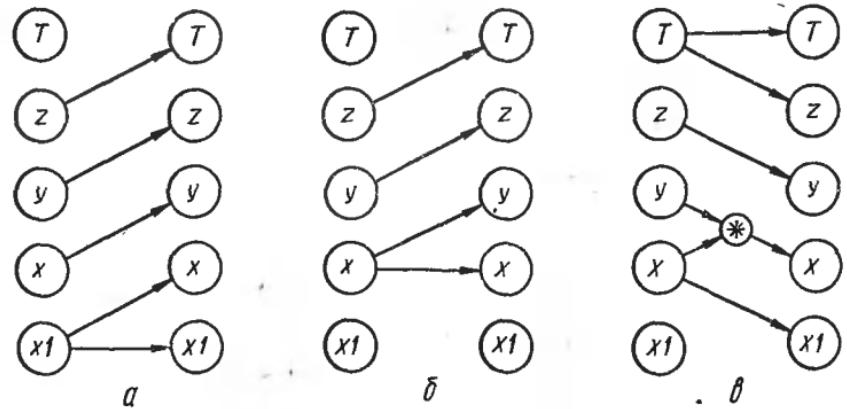


Рис. 30. Схема перемещения содержимого операционных регистров

При выполнении операторов косвенных переходов содержимое регистров $N = 0, 1, 2, 3$ с адресом условного перехода уменьшается на единицу, регистров $N = 4, 5, 6, 7$ увеличивается на единицу и остается неизменным для остальных регистров. Аналогично изменяется содержимое регистра N и для операторов косвенного обращения для записи и вызова содержимого регистров ЗУПВ.

Выполнение оператора BX приводит к перемещению операндов в регистрах стека и X1 в соответствии со схемой, показанной на рис. 30, а. Выполнение оператора засылки \uparrow в микрокалькуляторе рассматриваемого типа приводит к изменению содержания операционных регистров в соответствии со схемой, показанной на рис. 30, б.

Грамматические особенности выполнения двухместных операций можно отобразить схемой, показанной на рис. 30, в, где символом \times обозначена двухместная операция. Выполнение одноместных операций не изменяет содержимого регистров Y, Z и T, а содержимое регистра X (до выполнения операции) переписывается в регистр X1, тогда как в регистр X заносится результат выполненной операции.

Синтаксическим преимуществом входного языка микрокалькулятора рассматриваемого типа является автоматическое смещение операндов в регистрах операционного стека «вверх» не только при наборе числового операнда (после выполнения любой операции), но и при вызове в регистр X содержимого любого регистра ЗУПВ. Это исключает необходимость использования оператора \uparrow при вызове в регистр X содержимого регистров памяти и соответственно сокращает длину программы.

Несмотря на грамматические и лексические различия, входные языки программируемых микрокалькуляторов рассмотренных типов достаточно близки, и перевод программ с одного языка на другой не вызывает, как правило, значительных затруднений. При различном словарном запасе, как и для обычных языков, легче переводить со входного языка низшего уровня с более бедной лексикой, так как в противном случае отдельные операторы приходится отображать целыми предложениями. Так, перевод программ со входного языка микрокалькулятора типа «Электроника Б3-21» на входной язык микрокалькулятора типа «Электроника Б3-34» реализовать значительно легче, чем обратный перевод, хотя составить программу труднее для первого.

Для многих программ возможен дословный перевод, хотя он и не всегда оказывается наиболее кратким. Препятствием для дословного перевода, в частности, является наличие кольцевого стека у микрокалькулятора типа «Электроника Б3-21» (компенсируемого большей емкостью ЗУПВ у микрокалькулятора типа «Электроника Б3-34») и синтаксические различия в выполнении двухместных операций. Например, умножение константы, хранимой в регистре 8, на операнды, записанные в регистрах 2, 3, ..., N ЗУПВ, при необходимости записи результатов в те же регистры на входном языке микрокалькулятора типа «Электроника Б3-21» описывается предложением

$$F_8 \uparrow F_2 \times P_2 F_3 \times P_3 \dots F_N \times P_N.$$

Это предложение нельзя дословно перевести на входной язык микрокалькулятора типа «Электроника Б3-34» в связи с различиями в изменении содержимого операционного регистра Y. В этом случае необходимо восстанавливать константу после каждого умножения, что можно описать на входном языке микрокалькулятора типа «Электроника Б3-34» предложением

$$ИП8 ИП2 X \Leftrightarrow Y \times ЗП2 ИП3 BX \times ЗП3 \dots ИПN BX \times ЗПN.$$

При переводах программ, как и при переводах на обычных языках, переводчик должен достаточно хорошо знать особенности обоих языков. В связи с рассмотренными закономерностями перевода программ в дальнейшем в качестве основного будем использовать входной язык распространенного микрокалькулятора типа «Электроника Б3-21». Перевод программ, составленных на этом языке, на входные языки программируемых микрокалькуляторов других типов не представляет существенных затруднений.

3. СОСТАВЛЕНИЕ ПРОГРАММ

Текст программы для выбранного алгоритма вычислений обычно составляют в виде записываемой в строку или столбец последовательности символов операторов (команд). Иногда для начинающих программистов рекомендуют составлять программы в виде последовательности символов, обозначенных на нажимаемых клавишиах. Однако при наличии префиксных клавишей эти символы могут отличаться от символов вводимых функциональных операторов. Поэтому для облегчения записи и чтения программы целесообразно составлять ее из символов операторов независимо от способа их ввода в микрокалькулятор.

Запись операторов в строку или столбец компактна и достаточно удобна для введения в УЗП готовой программы. Однако при составлении разветвленных программ с большим числом операторов возникает необходимость в определении адресов операторов, которым передается управление. Поэтому бланк составляемой в строку или столбец программы приходится дополнять адресами каждого шага программы, что облегчает адресацию переходов и оценку оставшейся незаполненной емкости программной памяти.

Примером может служить табл. 41, в первой части которой приведены операторы программы (составленной по алгоритму, схема которого показана на рис. 22) на входном языке микрокалькулятора типа «Электроника Б3-21», а также их порядковые номера, адреса и, для сравнения, символы нажимаемых клавишей. Для определения символов операторов передачи управления при использова-

№ шага	1	2	3	4	5	6	7	8	9
Адрес	01	02	03	04	05	06	11	12	13
Оператор	F2	↑	F3	+	P2	1	0	×	P4
Клавиша	F 2	↑	F 3	+	P 2	1	0	×	P 4

№ шага	1	2	3	4	5	6	7	8	9
Адрес	00	01	02	03	04	95	06	07	08
Оператор	ИП2	ИП3	+	ЗП2	1	0	×	ЗП4	ИП2

ни подобной записи приходится обращаться к табл. 39. От этого недостатка свободны программы на языке микрокалькулятора типа «Электроника Б3-34» (вторая часть табл. 41), в котором вместо оператора передачи управления непосредственно указывается адрес перехода в виде десятичного двухзначного числа.

Операторы программы можно не дополнять их адресами при использовании двухмерной записи программы в прямоугольной системе координат. Для этого записываемая в столбец программа разбивается на части по шесть операторов (для микрокалькулятора «Электроника Б3-21»), каждая из которых записывается справа от предыдущей. В этом случае адрес $a_i b_i$ любого оператора определяется по номерам a_i столбца и b_i строки, на пересечении которых он находится.

Двухмерная запись программы удобна также для контроля правильности ее занесения в УЗП и для наглядного отображения переходов линиями, направленными от оператора передачи управления к оператору, которому передается управление. Например, приведенная в табл. 41 программа в двухмерной записи с указанием перехода имеет вид

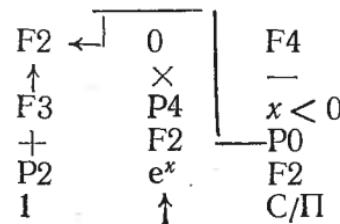


Таблица 41

10	11	12	13	14	15	16	17	18
14	15	16	21	22	23	24	25	26
F2	e^x	\uparrow	F4	-	$x < 0$	P0	F2	C/P
F 2	P \div	\uparrow	F 4	-	P БП	P 0	F 2	C/P

10	11	12	13	14	15	16	
09	10	11	12	13	14	15	
e^x	ИП4	-	$x < 0$	00	ИП2	C/P	

При такой записи легко определить, что, например, оператор e^x , расположенный на пересечении столбца с номером $a_i = 1$ и строки с номером $b_i = 5$ имеет адрес $a_i b_i = 15$.

Составление программ для микрокалькулятора «Электроника Б3-21» в двухмерной записи существенно упрощается при использовании трафарета с 60 прорезями, расположенными на пересечениях 10 столбцов и 6 строк. При составлении программы ее операторы последовательно вписывают в прорези очередного столбца, причем для операторов передачи управления оставляют свободную прорезь, которую заполняют после записи в последующей части программы оператора, которому передается управление. Если над каждой прорезью записать символы операторов передачи управления на соответствующий шаг программы, то отпадает необходимость в обращении к табл. 39. Прорези, соответствующие шагам программы, с которых нельзя обращаться к подпрограмме, целесообразно отметить рамками (рис. 31, а).

Для готовых и отлаженных программ удобна более компактная запись операторов в строки. Длинные программы в этом случае целесообразно разбивать на фрагменты по 6 (или по 10 для микрокалькулятора типа «Электроника Б3-34») операторов. Адрес $a_i b_i$ любого оператора при этом определяется номером a_i фрагмента и номером b_i оператора в этом фрагменте. При необходимости переходы можно указывать и при строчной записи программы*, например,

$|F2 \uparrow F3 + P2 1|0 \times P4 F2 e^x \uparrow |F4 - x < 0 P0 F2 C/\Pi|.$

Контроль правильности ввода подобных программ в УЗП облегчается при проверке по индикатору адресов $a_i 1$ первого или $a_i 6$ (в счетчике шагов в этом случае высвечивается число $a_i + 1; 0$) последнего операторов каждого 6-операторного фрагмента.

Составление программ в строчной записи упрощается при использовании трафарета (подобного показанному на рис. 31, б), в прорези которого операторы записывают по строкам. Номер фрагмента a_i легко определить по символам Ra_i операторов передачи управления первым операторам каждого фрагмента. Применение подобного трафарета избавляет от необходимости обращения к табл. 39 для кодирования адресов переходов.

При составлении программ в строчной записи для микрокалькулятора типа «Электроника Б3-34» трафарет не нужен, если в каждой строке записывать по 10 операторов, так как адрес любого оператора (и совпадающий с ним символ передачи управления этому оператору) легко определить по номеру $a_i = 0, 1, \dots, 9$ строки и номеру $b_i = 0, 1, \dots, 9$ оператора в этой строке.

Описание размещения исходных данных и результатов вычислений в регистрах памяти упрощается при обозначении содержимого индикаторного регистра X и операционного регистра Y символами

* В последующем изложении в качестве основного используется входной язык микрокалькулятора типа «Электроника Б3-21» и указания на тип входного языка будут приводиться только при использовании других входных языков.

PX и PY соответственно, а содержимого регистров ЗУПВ и стека (см. рис. 28) символами PN и CN , где N — номер регистра. При этом для описания введенных в память микрокалькулятора исходных данных целесообразно использовать формулы присвоения вида $a = PX$, $b = P2$, $c = C3$, где a , b и c — числа, заносимые в соот-

a_i	0	1	2	3	4	5	6	7	8	9
b_i	$P0$	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$	$P8$	$P9$
1	$F0$	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$F9$
2	$P\downarrow$	$PX \approx y$	PX	$P \div$	$P,$	$P/-$	PBP	PCx	$P-$	$P+$
3	0	1	2	3	4	5	6	7	8	9
4	$F\uparrow$	$FX \approx y$	FX	$F \div$	$F,$	$F/-$	FBP	FCy	$F-$	$F+$
5	\uparrow	$x \approx y$	X	\div	$,$	$I-I$	BPI	Cx	$-$	$+$
6										

a

$P0$	$F0$	$P\downarrow$	0	$F\uparrow$	\uparrow	$P1$	$F1$	$Px \approx y$	1	$Fx \approx y$	$x \approx y$
$P2$	$F2$	PX	2	FX	X	$P3$	$F3$	$P \div$	3	$F \div$	\div
$P4$	$F4$	$P,$	4	$F,$	$,$	$P5$	$F5$	$P/-$	5	$F/-$	$/-$
$P6$	$F6$	PBP	6	FBP	BPI	$P7$	$F7$	PCx	7	FCx	Cx
$P8$	$F8$	$P-$	8	$F-$	$-$	$P9$	$F9$	$P+$	9	$F+$	$+$

b

Рис. 31. Трафарет для двухмерной записи программ по столбцам (*a*) и по строкам (*b*)

ветствующие регистры памяти перед выполнением программы. Содержимое регистров памяти после выполнения программы (или, при необходимости, отдельных ее этапов) удобно описывать формулами вида $PX = f$, $P4 = g$, $C5 = h$, где f , g и h — результаты вычислений. Для сложных программ содержание регистров памяти на отдельных этапах вычисления удобно отображать таблично. Во всех

случаях, когда перед выполнением программы определенные регистры памяти должны быть очищены от прежнего содержимого, в описание исходных данных следует вводить соответствующие указания (например, $0 = P7, 0 = C1$ и т. п.).

Каждую составленную и отлаженную программу, предназначенную для многократного использования, целесообразно дополнять контрольным (тестовым) примером с указанием времени счета. Вычисления по этому примеру по введенной в УЗП программе позволяют проверить правильность занесения программы, а контрольные данные о времени ее выполнения позволяют оценить ожидаемое время решения задачи, что особенно существенно при использовании алгоритмов с замкнутыми циклами, когда возможно зацикливание программы.

При размещении исходных данных в регистрах памяти следует облегчать их запоминание и, следовательно, составление программы. Например, элементы квадратной матрицы размера 2×2 целесообразно размещать в регистры ЗУПВ в соответствии с расположением клавишей, нажимаемых для занесения исходных данных в эти регистры (см. рис. 27): $a_{11} = P7, a_{12} = P8, a_{21} = P4, a_{22} = P5$ или $a_{11} = P5, a_{12} = P6, a_{31} = P2, a_{22} = P3$. При занесении таких элементов в стек (см. рис. 28) их можно вводить по строкам (например, $a_{11} = C1, a_{12} = C2, a_{21} = C3, a_{22} = C4$) или по столбцам (например $a_{11} = C6, a_{21} = C5, a_{12} = C4, a_{22} = C3$).

Следует также учитывать удобство пользования микрокалькулятором, уменьшающее затраты времени при вычислениях. Например, часто заносимые при повторных пусках программы исходные данные целесообразно размещать в первую очередь в регистрах 8 и 7, клавиши номеров которых расположены близко к клавише Р. Аналогично вызываемые результаты вычислений целесообразно размещать прежде всего в регистрах 3, 2 и 6, клавиши номеров которых расположены близко к клавише F.

Исходные данные в виде целых чисел, отображаемых одной цифрой, целесообразно заносить непосредственно в программу, так как в этом случае ее длина не изменяется. Это справедливо, конечно, для случая, когда эти исходные данные представляют собой постоянные величины, не изменяемые при повторном использовании программы с новыми значениями аргументов.

Если в связи с недостаточным числом регистров памяти некоторые из исходных многозначных чисел приходится записывать непосредственно в программу, то следует учитывать, что каждый символ десятичного представления числа занимает один шаг программы. Например, занесение в программу постоянной Больцмана $k \approx 1,38 \cdot 10^{-23}$ занимает 8 шагов программы ($|1|, |3|, |8|, |B|, |P|, |2|, |3|, |-|, | |$).

Для сокращения числа шагов программы, занимаемых числовыми коэффициентами, следует использовать, когда это возможно, операторы обратной величины (например, $|5|, |1/x|$ вместо $|0|, |2|$), возведения в степень (например, $|5|, |2|x^2|$ вместо $|2|, |7|, |0|, |4|$), ком-

бинации операторов (например, $|4|x^2|1/x|$ вместо $|0|, |0|6|2|5|$), переход от десятичных к обычным дробям (например, $|7|\div|3|$ вместо $|2|, |3|3|3|3|3|3|$), округление чисел и т. п.

В тех случаях, когда программа предназначена для многократного использования после введения в УЗП (например, при вычислениях с изменяющимися исходными данными), следует предусмотреть все меры, облегчающие операции при повторных пусках программы и устраниющие последствия забывчивости пользователя. В частности, в подобных случаях очистку требуемых регистров памяти перед пуском программы целесообразно поручить самой программе. Для этого в начале программы, если она имеет длину меньше предельной, помещают фрагменты вида $0|PN|$ или $|0|\rightarrow|$, заносящие автоматически 0 в N -й регистр ЗУПВ или регистр С1 стека. Если перед повторными пусками программы должны быть очищены регистры, в которые заносились исходные данные или промежуточные результаты в процессе вычислений, то подобные фрагменты помещают в конце программы, когда содержимое соответствующих регистров не нужно (подобную очистку регистров, как показано ниже, можно реализовать и после оператора С/П, завершающего основную часть программы).

Результат вычислений удобно выводить на индикаторный регистр, но в тех случаях, когда результатом вычислений является несколько чисел, которые могут быть различными по характеру (например, вещественными или комплексными при вычислении корней уравнений), их целесообразно заносить программным путем в различные регистры памяти с выводом на индикатор номеров этих регистров. При этом, если длина программы меньше предельной, выводимые номера следует формировать так, чтобы они отличались от обычных представлений чисел (например, 08 для вещественного корня и 0405 для комплексного корня, вещественная и мнимая части которого занесены соответственно в регистры 4 и 5).

Если программой предусмотрено вычисление нескольких величин и одна из них выводится непосредственно на индикатор, то целесообразно обеспечить хранение последней и в одном из регистров ЗУПВ или стека, так как часто решение о регистрации полученного результата можно принять только после ознакомления со всеми полученными результатами, последовательно вызывая их в индикаторный регистр. Если первый из этих результатов дополнительно не занести в регистр памяти, то он будет стерт при вызове в регистр Х другого числа.

Поочередный вызов результатов, хранимых, например, в регистрах 6, 7 и 8 ЗУПВ и регистрах С1 и С2 стека, можно организовать, используя окончание программы вида $\dots|F6|C/P|F7|C/P|F8|C/P|\leftarrow|C/P|\leftarrow|C/P|$. В этом случае после вычислений по программе на индикаторе высвечивается содержимое регистра 6, но при дополнительных нажатиях клавиши С/П будут дополнительно вызываться числа, хранимые в соответствующих регистрах памяти. В тех случаях, когда для решения задачи используют несколько последова-

тельно выполняемых программ, окончание каждой из них удобно формировать так, чтобы после выполнения программы на индикаторе высвечивался номер следующей программы. Это особенно удобно, когда номер программы, используемой в качестве следующей, зависит от результата выполненных вычислений.

При составлении основной части программы необходимо рационально использовать все возможности входного языка, и, особенно, правила переноса содержимого регистров памяти для того, чтобы, по возможности, сократить длину программы и время вычислений.

Четкое представление об изменении содержания операционных регистров микрокалькулятора на каждом шаге выполняемой программы позволяет при ее составлении устраниТЬ избыточные операторы и уменьшить время счета. Например, вычисления по формуле (134) формально реализуются программой

$$F2 \ x^2 \uparrow F3 \times \uparrow F4 \times P5 \uparrow 1 + P6 \ F5 \uparrow P6 \div,$$

содержащей 17 операторов. Однако с учетом синтаксических особенностей используемого входного языка эту программу можно сократить на пять операторов: $F2 \ x^2 \uparrow F3 \times \uparrow F4 \times 1 + X \rightleftharpoons Y \div$. В последнем случае учтено, что вызов числа 1 автоматически переносит предыдущий результат $\omega^2 LC$ в регистр Y, а после выполнения операции сложения оба операнда остаются в операционных регистрах. ($PX = \omega^2 LC + 1$, $PY = \omega^2 LC$). Поэтому для выполнения операции деления достаточно поменять их местами.

Сохраняя в регистре Y исходный operand, можно сокращать длину программы и последовательным вводом двух двухместных операторов. Например, вычисление $y = 2a - b$ при $a = P2$, $b = P3$ выполнимо с помощью странного на первый взгляд фрагмента $|F2| \uparrow |F3| - | + |$, содержащего, однако, меньше операторов, чем прямая реализация $F2|2| \times | \uparrow |F3| - |$. Аналогично программа сокращается при вычислении $y = (a + b)b$ с помощью фрагмента $|F2| \uparrow |F3| + | \times |$ и в других подобных случаях.

При вычислениях функции одной переменной, предусматривающих многократное использование занесенной в УЗП программы при различных значениях аргумента, целесообразно начинать программу с оператора занесения аргумента (введенного в регистр X перед пуском программы) в один из регистров памяти и позаботиться о сохранности в регистрах памяти остальных исходных данных после выполнения программы.

В связи с ограниченной емкостью запоминающих устройств имеет значение рациональное распределение регистров памяти для записи исходных данных и результатов вычислений. Часто оказываются необходимыми буферные регистры для временного хранения operandов. Когда исходными данными заняты все регистры памяти, следует включать в вычислительный процесс те из исходных данных, которые не используются в дальнейшем, и использовать содержащие их регистры для записи промежуточных или окончательных результатов вычислений.

Используя в качестве буферных регистры стека, исходные данные целесообразно размещать таким образом, чтобы при переносе для хранения в стек содержимого регистра Х в последний одновременно вводился операнд, требуемый для выполнения последующей части программы. Например, часто встречающиеся в программах фрагменты вида ... |P2| F3|..., записывающие в регистр ЗУПВ результаты предыдущих вычислений и вызывающие операнды, хранимые в других регистрах, можно реализовать одним оператором \leftarrow или \rightarrow поворота стека, если в регистр С1 или С6 соответственно (см. рис. 28) предварительно занесен операнд, требуемый для вычислений в следующей части программы. При необходимости пропустить часть программы используется оператор безусловного перехода БП и оператор передачи управления по требуемому адресу.

Для начинающих программистов сложно составление разветвленных программ по алгоритмам, содержащим условные операторы, которые передают управление ветвям вычислительного процесса (начинающиеся операторами по адресам «Да» и «Нет»), содержащим функциональные блоки. В подобных случаях включение ветви, порождаемой передачей управления по адресу «Нет», в основную часть программы реализуется в простейшем случае с помощью оператора безусловного перехода БП и, при необходимости, временной записью операндов в буферные регистры памяти.

Пример 45. Составить программу вычисления функции $y = \operatorname{tg} \sqrt{\sin z/z}$.

При вычислении заданной функции необходимо отдельно учесть случай $z = 0$, для которого $\sin z/z = 1$. Следовательно, вычисление заданной функции можно описать следующим алгоритмом:

1. Если $z = 0$, то перейти к п. 2, иначе к п. 3.
2. Принять $x = 1$ и перейти к п. 4.
3. Вычислить $x = \sqrt{\sin z/z}$ и перейти к п. 4.
4. Вычислить $\operatorname{tg} x$.

В соответствии со схемой этого алгоритма (рис. 32) условный оператор при выполнении условия $z = 0$ передает управление блоку 2, присваивающему текущей переменной значение 1 и передающему его в блок 3, который вычисляет $\operatorname{tg} 1 = 1,5574074$. Если условие не выполняется, то управление передается блоку 4, вычисляющему текущее значение переменной $x = \sqrt{\sin z/z}$ и передающему его блоку 3.

Вычисления в блоке 4 формально реализуются фрагментом $|F2| \sin |↑| F2 |÷| \sqrt{| } |$. Однако при использовании грамматических особенностей языка целесообразно заменить его фрагментом $|↑| \sin |X \leftarrow Y |÷| \sqrt{| } |$, содержащим один оператор меньше. При вычислении $\operatorname{tg} x$ операторами $e^{jx} |÷|$ решение задачи после занесения исходного числа z в индикаторный регистр реализуется программой

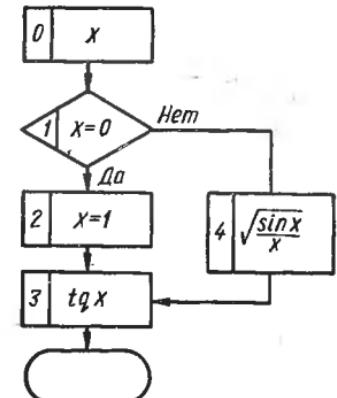


Рис. 32. Схема алгоритма вычисления функции

$$x = 0 \quad P1 \quad 1 \quad e^{jx} \div C/P \quad \uparrow \quad \sin X \leftarrow Y \div \sqrt{| } \quad e^{jx} \div C/P.$$

В этой программе обе ветви, побуждаемые условным оператором, не соединяются до конца вычислений.

Для более сложных программ подобное решение непригодно в связи с большим числом требуемых операторов, и обе ветви обычно соединяют с их общей частью при помощи оператора БП безусловного перехода. В рассматриваемой задаче это приводит к программе

$$x = 0 \uparrow 1 \text{ БП } FX \geq Y \uparrow \sin X \geq Y \div V^- e^{jx} \div C/P.$$

Эта более короткая программа на входном языке микрокалькулятора типа «Электроника Б3-34» записывается как

$$x = 0 \uparrow 05 1 \text{ БП } 10 \uparrow \sin X \geq Y \div V^- \downarrow \text{tg } C/P.$$

Составленные программы пригодны для значений $\sin z/z > 0$. Если требуется вычислять заданную функцию за пределами этой области, то после оператора « \div » с адресом 13 (для микрокалькулятора типа «Электроника Б3-21») можно ввести оператор смены знака с помощью фрагмента $|x < 0| A| -/|$, где A — оператор передачи управления команде, следующей за этим фрагментом.

Оператор безусловного перехода БП целесообразно также использовать после оператора остановки вычислений С/П в тех случаях, когда программа предназначена для многократного использования при различных исходных данных. Окончание программы в этих случаях записывается как ... | С/П | БП | А | ... , где A — оператор передачи управления по адресу функционального оператора, расположенного в основной части программы. Подобное окончание программы устраняет необходимость в нажатии клавишей БП и кодирующих адрес перехода, если выполнение программы начинается не с начала, или клавиши В/О, если при повторном пуске выполнение программы начинается с начального шага. В этих случаях для повторного пуска программы достаточно нажать только клавишу С/П. Например, при вычислении дисперсии случайной величины для накопления суммы квадратов ее значений целесообразно воспользоваться программой

$$|x^2| \uparrow |F2| + |P2| C/P | БП | P0|.$$

При первом вводе случайной величины в регистр Х программа выполняется нажатием клавиши В/О и С/П, но при последующих вводах — нажатием только клавиши С/П.

Часто вычислительный процесс содержит операции, не связанные с изменяющимися перед каждым пуском программы исходными данными. Для ускорения вычислений при повторных пусках программы, реализующей подобные вычисления, после оператора С/П записывают БП и оператор передачи управления по адресу, с которого начинаются вычисления над изменяемыми перед каждым пуском программы исходными данными. При этом необходимо предусмотреть запоминание результата предыдущих вычислений (выполненных при первом исполнении программы) для его использования в повторно выполняемой части программы.

Многократное обращение к программе, занесенной в УЗП, характерно при исследовании функциональных зависимостей. В этом случае для удобства вычислений и сокращения времени подготовки данных перед следующим пуском программы целесообразно предусмотреть автоматическую подготовку таких данных, что также можно реализовать с помощью оператора БП в конце программы. Например, при занесении начального значения аргумента x_0 и его постоянного приращения Δx соответственно в регистры 2 и 3 ЗУПВ программу целесообразно дополнить окончанием... |С/П|F2|
↑|F3|+|P2|БП|P0|. Тогда при повторном пуске программы нажатием только клавиши С/П вначале будет вычислено новое значение аргумента $x_{i+1} = x_i + \Delta x$, а затем начнутся вычисления по основной программе с новым значением аргумента. В этом случае вычисления сводятся к повторным нажатиям клавиши С/П и к занесению полученных результатов в вычислительный бланк.

Включение оператора БП в окончание программы целесообразно использовать и для восстановления исходных данных в регистрах памяти, используемых в процессе вычислений в качестве буферных для временного занесения промежуточных результатов или регистров, в которые занесены окончательные результаты. Например, при необходимости очистки заданных регистров перед повторными пусками программы окончание записывается как ... |С/П|0|РN|БП|P0|, где РN обозначает один или несколько операторов, заносящих 0 в регистры ЗУПВ. Аналогично, хотя и несколько большим числом операторов, реализуется очистка нескольких регистров стека перед повторным пуском программы. Подобная предусмотрительность сокращает время вычислений и устраняет ошибки, связанные с рассеянностью пользователя при повторных выполнениях программы без выключения микрокалькулятора (когда содержимое всех регистров памяти и УЗП стирается).

В качестве оператора безусловного перехода можно использовать также оператор В/О, записываемый в конце программы после оператора С/П. В этом случае при повторном пуске программы нажатием клавиши С/П ее выполнение начинается с адреса 02. Использование в программах операторов условного перехода существенно расширяет возможности программирования, обеспечивая, в частности, формирование замкнутых циклов, в которых вычисления продолжаются до выполнения некоторого условия.

Условием окончания вычислений в цикле может быть заданное число этих вычислений (например, при суммировании заданного числа членов ряда). Для подсчета числа выполненных циклов при подобных вычислениях программа должна содержать счетчики. Простейший счетчик реализуется фрагментом вида ... |F2|1|+|P2|..., где после каждого выполнения части программы, в которой расположен счетчик, содержимое регистра 2 (или, соответственно, любого другого регистра), равное нулю перед началом вычислений, увеличивается на единицу. Подобные счетчики можно использовать и в многократно выполняемой программе для подсчета числа ее повтор-

ных пусков. В последнем случае часто содержимое счетчика удобно выводить на индикатор для непосредственного контроля.

Достижение заданного числа повторных вычислений (занесенного, например, в регистр 3) контролируется условным оператором, располагаемым после счетчика в фрагментах вида

$$\dots |F2|1| + |P2|\uparrow|F3|-|x=0|\dots$$

Если заданное число повторных выполнений программы или той ее части, в которой расположен счетчик, достигнуто, условный оператор $x = 0$ передает управление оператору, обеспечивающему выполнение дальнейших вычислений. Длину счетчика и число используемых в нем регистров можно сократить, если записать его в виде $\dots|F3|1|-|P3|x=0|\dots$, где в регистр 3 при каждом выполнении программы необходимо записать заданное число повторных вычислений.

При использовании счетчиков в замкнутых циклах условный оператор прекращает вычисления в цикле после заданного числа их повторений, которое можно определить не заранее заданным числом, а условием задачи.

Пример 46. Составить программу вычисления факториала $n!$.

Вычисление факториала можно свести к последовательности вычислений произведений $n(n - 1); n(n - 1)(n - 2); \dots$ до тех пор, пока последний сомножитель не станет равным единице. При этом нужно учесть и случай $n = 0$ с факториалом $0! = 1$.

В последнем случае выполнение условия $n = 0$ для текущей переменной порождает весьма короткую ветвь. Поэтому можно сократить длину программы без использования оператора БП, если индентифицировать нулевое значение заданного числа n оператором $x \neq 0$. Тогда решение задачи описывается следующим алгоритмом:

1. Принять $n_0 = n, k_0 = 1$ и перейти к п. 2.
2. Если $n_0 \neq 0$, то перейти к п. 3, иначе к п. 6.
3. Вычислить $k_{i+1} = n_i k_i$ и перейти к п. 4.
4. Вычислить $n_{i+1} = n_i - 1$ и перейти к п. 5.
5. Если $n_{i+1} = 0$, то перейти к п. 6, иначе к п. 3.
6. Принять $n! = k_{i+1}$ и закончить вычисления.

Схема этого алгоритма показана на рис. 33. Для его программной реализации вначале введем автоматическое занесение заданного числа из регистра X в регистр 3. Тогда пункт 1 описания алгоритма реализуется при занесении единицы в регистр 2. После этого условный оператор $x \neq 0$ проверяет содержимое регистра 3 и при невыполнении условия вызывает в индикаторный регистр содержимое регистра 2, равное $1 = 0!$ При выполнении условия управление передается по адресу «Да» оператору, заносящему число n в регистр 2, после чего вычисляются значения $k_{i+1} = k_i n_i$ и $n_{i+1} = n_i - 1$. Если результат последней операции не равен нулю, то значение k_{i+1} передается в регистр 2 и вычисления продолжаются. В противном случае значение $k_{i+1} = n!$ высвечивается на индикаторе и вычисления прекращаются. Описанные операции автоматически выполняются по программе 1.

Программа 1. Вычисление факториала числа

P3	1	P2	F3	$x \neq 0$	$F \times$	P3	\uparrow	F2	\times	P2	F3
1	-	$x=0$	P1	F2	C/P						

При вычислениях по программе 1* можно руководствоваться следующими контрольными данными: время вычисления $10! = 3628800$ составляет около 15 с, время вычисления $20! = 2,432901 \cdot 10^{18}$ — около 35 с.

Программу 1 можно «дословно» перевести на входной язык микрокалькулятора типа «Электроника Б3-34» как

```
ЗП3 1 ЗП2 ИП3  $x \neq 0$  16 ЗП3 ↑ ИП2 × ЗП2 ИП3  
1 —  $x=0$  06 ИП2 С/П.
```

Однако с учетом синтаксических особенностей этого входного языка в таком переводе избытен оператор \uparrow по адресу 07, а использование оператора-счетчика L2 дополнительно сокращает перевод:

```
 $x=0$  03 1 ЗП2 ЗП3 L2 09 ИП3 С/П ИП3 ИП2 × БП  
04.
```

В связи с ограниченной разрядностью микрокалькулятора факториал вычисляется по программе, составленной в предыдущем примере, с точностью не более восьми значащих цифр. Аналогичная ситуация характерна и при использовании универсальных ЭВМ, среди которых только ЭВМ серии МИР без использования специальных приемов обеспечивают вычисления с точностью до 999 значащих цифр.

При составлении программ с итерационными циклами и суммированием бесконечных рядов большое значение имеет правильный выбор критерия автоматического прекращения вычислений, причем для программируемых микрокалькуляторов с ограниченной емкостью УЗП важна простота программной реализации этого критерия.

Простейшее условие прекращения вычислений при монотонном стремлении переменной x к некоторому пределу заключается в достижении приращением переменной предельного значения

$$x_{i+1} - x_i < \varepsilon \quad (136a)$$

при монотонном увеличении x или

$$x_i - x_{i+1} < \varepsilon \quad (136b)$$

при монотонном уменьшении x , где ε — наперед заданная малая величина.

Если приращение переменной после очередной итерации может изменяться по знаку, то в качестве критерия прекращения вычислений проще всего принять условие

$$(x_{i+1} - x_i)^2 < \varepsilon. \quad (137)$$

* Программы, имеющие самостоятельное значение, обозначены в книге порядковыми номерами. Номера программ на входном языке микрокалькулятора типа «Электроника Б3-34» дополнены цифрами 34 через косую черту.

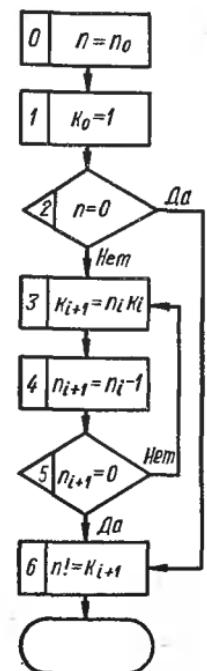


Рис. 33. Схема алгоритма вычисления факториала

Для сокращения длины программы в качестве величины ε можно выбирать машинный нуль, что соответствует условию прекращения вычислений

$$x_{i+1} - x_i = 0, \quad (138)$$

при котором достигается максимальная точность при фиксированной разрядности. Однако возможность использования критерия (138) зависит как от характера решаемой задачи, так и от выбранного алгоритма ее решения.

Пусть для решения уравнения $y = f(x) = 0$ используется метод касательных Ньютона. Тогда в связи с конечной разрядностью операндов и их округления в процессе вычисления может сложиться ситуация, при которой, начиная с некоторой итерации, переменные x и y будут циклически изменяться вблизи искомого корня, не достигая значения машинного нуля, и программа зацикливается. Например, при решении уравнения $x^3 - 0,03x^2 + 0,03x - 0,01000002 = 0$ на 8-разрядном микрокалькуляторе типа «Электроника Б3-21», методом касательных при приближении к искомому корню $x_k = 0,17774429988$ «справа» образуется следующий бесконечный цикл:

$$\begin{aligned} x_i &= 0,17774431; y_i = 10^{-9} \text{ (вместо более точного } 1,16 \cdot 10^{-9}); \\ x_{i+1} &= 0,17774430; y_{i+1} = -10^{-9} \text{ (вместо более точного } 1,6 \cdot 10^{-11}); \\ x_{i+2} &= 0,17774431; y_{i+2} = 10^{-9} \text{ и т. д.} \end{aligned}$$

При таких «блужданиях» вблизи корня x_k ни для приращения переменной x , ни для невязки уравнения y значение машинного нуля никогда не будет достигнуто. Поэтому для выбранного алгоритма решения уравнения критерий максимальной точности (138) не гарантирует окончания итерационного процесса и приходится обращаться к критерию (137), гарантирующему окончание вычислений при выборе ε больше машинного нуля.

При использовании 8-разрядного микрокалькулятора модули абсолютных приращений связанных между собой аргумента x и невязки или значения функции y подчиняются условиям

$$\Delta x_{\min} = |x_k| \cdot 10^{-7}; \quad (139a)$$

$$\Delta y_{\min} = |f'(x_k)| \cdot x_k \cdot 10^{-7}, \quad (139b)$$

так как в противном случае их значение попадает в интервал машинного нуля. Поэтому для получения высокой точности результата величину ε в неравенствах вида (136) или (137) следует выбирать близкой к минимальному приращению контролируемой переменной, для чего необходимо знать приближенное значение искомого корня при использовании критерия (139a).

При использовании неравенства (139b) требуются предварительные сведения о значении производной $f'(x)$ вблизи искомого корня. Поэтому вместо критерия абсолютного приращения вида (137) при контролировании приращения аргумента целесообразно переходить к оценке относительного приращения переменной $\Delta x_i/x_i$, обеспечи-

вающей окончание вычислений на 8-разрядном микрокалькуляторе при

$$\left(\frac{x_{l+1} - x_l}{x_l} \right)^2 \leqslant 10^{-14}. \quad (139\text{в})$$

Проверка относительного приращения невязки y вблизи корня не имеет смысла, так как эта величина не стремится к нулю.

При решении уравнений вида $y = f(x) = 0$ методом половинного деления интервала, в котором находится корень, в соответствии с отношением $\Delta x_i = \Delta x_0 / 2^i$, где Δx_0 — начальная длина интервала, изменение Δx_i независимо от остальных изменяющихся переменных. Поэтому критерий максимальной точности (138) гарантирует окончание итерационного процесса при любом виде решаемого уравнения, а также при суммировании сходящихся бесконечных рядов или вычислении несобственных интегралов численными методами. В этом случае критерий (138) принимает вид

$$\sum_{k=0}^{N+1} A_k - \sum_{k=0}^N A_k = 0 \text{ или } \sum_{k=0}^N A_k + A_{N+1} = \sum_{k=0}^N A_k. \quad (140)$$

Программная реализация этого критерия (см. приложение), например, при суммировании сходящегося ряда после вычисления очередного члена, обеспечивается фрагментом вида... ↑ F4 X ⇌ Y + P4 — x = 0 A F4 C/P. В этом случае в регистре 4 накапливается сумма вычисленных членов ряда и, если $\sum_{k=0}^{i+1} A_k - \sum_{k=0}^i A_k \neq 0$, то управление передается оператором A предыдущей части программы, вычисляющей значение следующего члена ряда.

Пример 47. Составить программу для вычисления суммы членов ряда (геометрической прогрессии)

$$y = \sum_{k=0}^{\infty} qx^k = q(1 + x + x^2 + x^3 + \dots),$$

сходящегося при $|x| < 1$.

Очередной член заданного в скобках ряда равен произведению предыдущего члена на значение аргумента. Следовательно, сумма членов ряда вычисляется сложением очередного члена ряда с предыдущей суммой до выполнения критерия максимальной точности (140). Обозначив очередной член ряда и текущую сумму членов символами a_l и S_l соответственно, опишем решение задачи следующим алгоритмом:

1. Принять $a_0 = 1$, $s_0 = 1$ и перейти к п. 2.
2. Вычислить $a_{l+1} = a_l x$ и перейти к п. 3.
3. Вычислить $S_{l+1} = S_l + a_{l+1}$ и перейти к п. 4.
4. Если $S_{l+1} - S_l = 0$, то перейти к п. 5, иначе к п. 2.
5. Вычислить $y = S_l q$ и закончить вычисления.

Составленный алгоритм реализуется программой 2, перед пуском которой исходные данные q и x заносятся соответственно в регистры 7 и X. Результат вычислений высвечивается на индикаторе.

Программа 2. Вычисление геометрической прогрессии

P8 1 P2 P3 F2 ↑ F8 × P2 ↑ F3 X ⇌ Y + P3 — x = 0 F↑ F3
↑ F7 × C/P.

При $q = 1$ и $x = 0,1$ время вычислений $y = 1,111111$ составляет около 16 с, при $q = 1$ и $x = 0,5$ время вычисления $y = 2$ возрастает до 55 с. При увеличении аргумента до $x = 0,9$ время вычислений увеличивается до 6 мин.

Отредактированный перевод программы 2 на входной язык микрокалькулятора типа «Электроника Б3-34» содержит на три оператора меньше

ЗП8 1 ЗП2 ЗП3 ИП2 ИП8 \times ЗП2 ИП3 + ЗП3 ВХ — $x = 0$ 04
ИП3 ИП7 \times С/П.

Для суммирования медленно сходящихся или расходящихся рядов целесообразно составлять программы, вычисляющие сумму заданного числа членов ряда. Для этого в цикл включается счетчик числа членов и условный оператор, прекращающий вычисления после сложения заданного числа членов. Для медленно сходящихся рядов достаточно оценить разность двух очередных значений суммы членов ряда. Кажущаяся возможность упрощения критерии (137) или (140) заменой их условиями $\Delta x = 0$ (если Δx вычисляется независимо) или $A_{N+1} = 0$ соответственно обычно нецелесообразна, так как в этом случае вычисления прекратятся только после достижения абсолютного машинного нуля $\Delta x < A_{\min} = 10^{-99}$, что чрезмерно увеличит время вычислений.

При неоднократном использовании однотипных фрагментов программы может оказаться целесообразным их вынесение в подпрограмму, помещаемую после оператора С/П, завершающего основную часть программы. Если подпрограмм несколько, то их записывают последовательно и каждую оканчивают оператором В/О. Обращение к любому оператору подпрограммы реализуется оператором ПП и оператором передачи управления по соответствующему адресу. В самой подпрограмме можно записать обращение к другой подпрограмме.

Целесообразность введения подпрограммы можно оценить следующим образом. Пусть в программе содержится фрагмент, занимающий m шагов и повторяющийся n раз. Его непосредственная запись занимает mn шагов. Выделение этого фрагмента в подпрограмму потребует (кроме m шагов, занимаемых основной частью подпрограммы) $2n$ шагов для записи оператора ПП и операторов передачи управления, а также одного шага для записи оператора В/О. Следовательно, введение подпрограммы сокращает длину программы (иногда увеличивая время счета) при выполнении условия $(2n + m + 1) < mn$ или $m > (2n + 1)/(n - 1)$. Для числа обращений $n = 2$ должно быть $m > 5$, при $n = 3$ должно быть $m > 4$ и т. д.

Пример 48. Составить программу вычисления алгебраического многочлена пятой степени.

В соответствии с выражением (33) заданный многочлен целесообразно вычислять по формуле

$$y = (((a_5x + a_4)x + a_3)x + a_2)x + a_1)x + a_0,$$

которую можно рассматривать как компактную запись алгоритма решения. Размещая исходные данные в ЗУПВ как $a_5 = P2$, $a_4 = P3$, $a_3 = P4$, $a_2 =$

= P5, $a_1 = P6$, $a_0 = P7$, $x = PX$, нетрудно составить следующую программу:

| P8 ↑ F2 × ↑ F3 | + ↑ F8 × ↑ F4 | + ↑ F8 × ↑ F5 |
+ ↑ F8 × ↑ F6 | + ↑ F8 × ↑ F7 | + C/P БП P0 |.

В этой программе четыре фрагмента $| + \uparrow F8 \times \uparrow |$ идентичны, а один отличается от них лишь одним оператором. Следовательно, программу можно сократить, вынося повторяющийся фрагмент в подпрограмму. В этом случае получим программу 3, содержащую на семь операторов меньше исходной.

Программа 3. Вычисление алгебраического многочлена степени 5

P8 F2 ПП F ÷ F3 ПП| 3 F4 ПП 3 F5 ПП| 3 F6 ПП 3 F7 + |
C/P БП P0 + ↑ F8 | × ↑ B/O |.

Вычисление многочлена $y(1) = 2x^5 + 3x^4 + 4x^3 + 5x^2 + 6x + 7 = 27$ по первой из составленных программ выполняется около 5 с, по программе 3 — около 8 с. При $x = 0$ время счета около 12 с, что характерно для автоматического вычисления многих функций при нулевом значении аргумента.

Программа для микрокалькулятора типа «Электроника Б3-34» при использовании оператора косвенного обращения по содержимому регистра 0, уменьшающему при каждом обращении на единицу, содержит ($a_5 = P6$, $a_4 = P5$, $a_3 = P4$, $a_2 = P3$, $a_1 = P2$, $a_0 = P1$, $x = PX$) всего 19 операторов:

| ЗП7 6 ЗП0 ИП6 ↑ X \leftarrow Y ИП7 × КИПО + | ИП0 1 -- x = 0 05
X \leftarrow Y C/P БП 00 |.

В начале программы в регистр-счетчик заносится число 6, обеспечивающее при первом использовании этого оператора вызов операнда из регистра 5 и 5-кратное повторение цикла в дальнейшем.

В рассмотренном примере введение подпрограммы несколько увеличило время счета, но уменьшило длину программы примерно на 25 %. Подобное сокращение длины может оказаться необходимым, если программа вычислений не помещается в УЗП.

4. ОПТИМИЗАЦИЯ ПРОГРАММЫ

Автоматизация решения сложных задач с помощью программируемых микрокалькуляторов затрудняется в связи с малой емкостью их запоминающих устройств. Поэтому решение таких задач часто приходится разбивать на ряд этапов, реализуемых несколькими последовательно вводимыми в УЗП программами. При большом числе исходных данных может оказаться возможным лишь автоматизация собственно вычислений при вводе перед каждым пуском программы очередных operandов. Такой вынужденный подход к автоматизации вычислений приводит к относительно большим затратам времени для ввода программ или исходных данных. Вместе с тем даже в тех случаях, когда программа вычислений и исходные данные помещаются в запоминающих устройствах, время выполнения программы может оказаться чрезмерно большим. Поэтому, в конечном итоге, основным критерием качества (оптимальности) программы является не ее длина, а затраты времени на решение задачи с помощью этой программы, включая ввод программы и исходных

данных, регистрацию в вычислительном бланке промежуточных результатов и т. д.

Решение одной и той же задачи можно автоматизировать частично или полностью с помощью различных программ, отличающихся как способом программной реализации одного и того же алгоритма, так и выбором алгоритма решения задачи, что часто связано с изменением математической формулировки задачи. Следовательно, задача оптимизации программы по критерию минимальных затрат времени на решение задачи при полном использовании особенностей входного языка достигается только при надлежащем выборе алгоритма решения и математической формулировки вычислительной задачи.

Одним из простейших приемов оптимизации программ является нормирование исходных данных и результатов операций. Нормирование численных значений параметров математических моделей, имеющее существенное значение и при вычислениях в обычном режиме, позволяет ускорить решение задачи в программируемом режиме и уменьшить число регистров, требуемых для размещения исходных данных, увеличивая тем самым размеры решаемой задачи. Нормирование исходных данных с одинаковой размерностью целесообразно выполнять таким образом, чтобы часть из них оказалась равными единице или, по крайней мере, однозначными числами. Это дает возможность ввести часть исходных данных непосредственно в программу без увеличения ее длины и освободит часть регистров памяти. Кроме того, нормирование однотипных параметров уменьшает и время счета.

Более эффективным может оказаться нормирование коэффициентов и переменных конкретной математической модели, связывающей исходные данные с искомыми результатами вычислений. Например, числитель и знаменатель дробно-рациональной функции вида (бб) можно разделить на один из ее отличных от нуля коэффициентов a_i , что не изменит величины функции, но уменьшит по крайней мере на два оператора длину программы (так как вместо $a_i x^i$ вычисляется x^i), сократит время вычислений и освободит регистр памяти. Если числитель и знаменатель дробно-рациональной функции разделить соответственно на один из их коэффициентов, то для вычисления нормированного значения функции потребуется на два регистра памяти и, по крайней мере, на четыре оператора меньше, чем для вычисления исходной функции.

Нормирование целесообразно использовать и для трансцендентных выражений. Например, приняв в выражении

$$y = A(e^{\Lambda x} - 1) + Cx$$

подстановки $w = y/A$ и $z = \Lambda x$, можно преобразовать его в равенство

$$w = e^z + az - 1,$$

содержащее только один коэффициент $a = c/A\Lambda$. Вычисление такого выражения требует значительно меньших затрат времени, а в тех случаях, когда y задано и требуется решить подобное уравнение относительно x , итерационный процесс заканчивается быстрее. После вычисления несложно определить y и x обратными подстановками. Часто последние операции излишни, так как при исследовании характера зависимости $y(x)$ замена ее зависимостью $w(z)$ приводит лишь к изменению масштаба.

Другой путь сокращения программы заключается в такой организации вычислений, при которой число операций по вызову и размещению operandов в регистрах памяти минимально. В качестве примера рассмотрим вычисление функции

$$y = (ab - cd)/(fb - d^2)$$

при записи исходных данных $a = P2$, $b = P3$, $c = P4$, $d = P5$, $f = P6$. Программа вычислений, составленная непосредственно по заданной формуле, будет иметь примерно такой вид:

$$\begin{aligned} |F6 \uparrow F3 \times \uparrow F5| x^2 - P7 F4 \uparrow F5 | \times P8 F2 \uparrow F3 \times | \uparrow F8 \\ - \uparrow F7 \div |C/\Pi|. \end{aligned}$$

Если делением на b привести заданную формулу к виду

$$y = (a - cd/b)/(f - d^2/b),$$

то при том же размещении исходных данных программа вычислений будет содержать на четыре оператора меньше:

$$|F5 \uparrow F3 \div P7 \times | \uparrow F6 - P8 F7 \uparrow |F4 \times \uparrow F2 - \uparrow |F8 \div C/\Pi.$$

Следовательно, длина программы уменьшилась на 16%, что может оказаться существенным при многократных вычислениях по заданной формуле (время счета при этом сокращается примерно на 20%).

Особенно значительный выигрыш в длине программы и времени счета можно получить при удачном выборе алгоритма решения задачи. Проблема оптимального выбора алгоритмов в общем случае не решена, но внимательное изучение условий задачи, использование эвристических приемов, опыта и даже интуиции программиста (учитывая, что «интуиция — дитя информации») может привести к успешному результату такого выбора.

Рассмотрим в качестве примера составление программы вычисления числа сочетаний из n по m элементов, определяемого формулой

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Для решения этой задачи проще всего воспользоваться программой 1 вычисления факториала, выделив ее в подпрограмму, последовательно вычисляющую $n!$, $m!$ и $(n-m)!$. В этом случае решение

рассматриваемой задачи при $n = P2$, $m = P3$ выполняется программой

F3	PП	P \div	P8	F2	PП	P \div	F8	\div	P8	F2	\uparrow
F3	—	ПП	P \div	F8	XY	\div	C/P	P4	1	P5	F4
$x \neq 0$	P6	P4	\uparrow	F5	\times	P5	F4	1	—	$x = 0$	\rightarrow
F5	\uparrow	B/O	.								

Для сокращения длины составленной программы представим исходную формулу в виде

$$C_n^m = \frac{\prod_{k=0}^{m-1} (n-k)}{m!}.$$

Факториал $b!$ при $a = 1$ является частным случаем выражения

$$N = \prod_{k=a}^b k.$$

Исключив случай $b = 0$ с очевидным результатом $C = 1$, составим для вычисления этого выражения при $a = P6$, $b = P5$ подпрограмму $|F4 \times P4 F5 1 - |x \neq 0 A_1 P5 \uparrow F6 - |x = 0 A_2 F4 B/O|$. Вычисления по этой подпрограмме начинаются с ее четвертого шага (оператор F5), оператор A_2 передает управление первому шагу, а оператор A_1 — оператору F4, вызывающему результат на индикатор в конце подпрограммы.

Сохраняя размещение исходных данных, составим программу 4 с использованием рассмотренной подпрограммы.

Программа 4. Определение числа сочетаний из n по m элементов

F3	P4	P5	1	P6	PП	P4	P8	F2	P4	P5	\uparrow
F3	—	P6	ПП	P4	\uparrow	F8	\div	C/P	F4	\times	P4
F5	1	—	$x \neq 0$	/—/	P5	\uparrow	F6	—	$x = 0$	3	F4
B/O											

В этой программе первые пять операторов готовят исходные данные для обращения к подпрограмме по адресу 41, вычисляющей значение $m!$ при $a = 1$, $b = m$. Затем по командам... $|F2 \dots P6| \dots$ в регистры 4 и 5 записывается значение n , а в регистр 6 величина $(n - m)$. При втором обращении к подпрограмме вычисляется произведение

$$\prod_{k=n-m+1}^m k = \prod_{k=0}^{m-1} (n - k),$$

которое делится на вычисленное ранее значение $m!$ с выводом результата на индикатор. Содержание регистров ЗУПВ на различных этапах вычислений по этой программе иллюстрируется данными табл. 42.

Таблица 42

PN	Перед пуском программы	Перед первым обращением к ПП	После первого обращения к ПП	Перед вторым обращением к ПП	После второго обращения к ПП	После выполнения программы
2	n	n	n	n	n	n
3	m	m	m	m	m	m
4	—	m	$m!$	n	$\prod_{k=0}^{m-1} (n-k)$	$\prod_{k=0}^{m-1} (n-k)$
5	—	m	1	n	$n-m+1$	$n-m+1$
6	—	1	1	$n-n$	$n-m$	$n-m$
7	—	—	—	—	—	—
8	—	—	—	$m!$	$\prod_{k=0}^{m-1} m!$	$m!$
X	—	1	$m!$	$n-m$	$\prod_{k=0}^{m-1} (n-k)$	C_n^m

Таблица 43

Исходные данные	m	3	1	4	1	5
	n	5	9	9	15	15
Исходная программа	C_n^m	10	9	126	14,99999	3002,999
	$t, \text{ с}$	23	23	33	50	50
Программа 4	C_n^m	10	9	126	15	3003
	$t, \text{ с}$	13	7	17	7	21

Результаты вычислений по исходной программе и программе 4 приведены в табл. 43 для нескольких значений n и m . Сравнение данных табл. 43 свидетельствует об удачном выборе алгоритма для программы 4, позволившем уменьшить ее длину и время вычислений. Для тривиальных случаев $m = n$, $m = 0$ и $n = 0$ программу 4 можно дополнить проверкой условий $m = 0$ и $m - n = 0$ в начале программы.

Исходные данные инженерных задач обычно известны лишь приближенно. Поэтому целесообразно округлять их до соответствующего числа значащих цифр и, с учетом реальной точности результатов вычислений, округлять коэффициенты расчетных формул, что сократит длину программ и время счета.

Пример 49. Составить программу вычисления вторичных параметров коаксиальных линий передачи (Z_0 , β , α) по заданной геометрии (внутреннему d и внешнему D диаметрам), характеристикам заполняющего линию диэлектрика (μ_r , ϵ_r , $\operatorname{tg} \delta$) и проводников (μ_{pr} , σ , ρ) для заданной рабочей частоты f .

Расчетные соотношения для вычисления искомых параметров имеют вид

$$Z_0 \approx V L' / C'; \quad \beta \approx \omega V L' C' \quad \alpha \approx R' / 2Z_0 + g' Z_0 / 2,$$

где погонные параметры линии

$$L' \approx \frac{\mu}{2\pi} \ln(D/d); \quad C' = \frac{2\pi\epsilon}{\ln(d/D)}; \quad R' \approx \left(\frac{1}{d} + \frac{1}{D} \right) \sqrt{\frac{\mu_0 f}{\pi\sigma}} + \frac{4\rho}{\pi d^2}; \quad g' = 2\pi f C' \operatorname{tg} \delta,$$

а относительные значения μ и ϵ связаны с абсолютными соотношениями

$$\mu = 4\pi \cdot 10^{-7} \mu_r; \quad \epsilon = 10^{-9} \epsilon_r / 36\pi,$$

причем обычно $\mu_r = 1$. Если проводники изготовлены из меди, то с учетом малой точности определения α расчетные соотношения после округления коэффициентов можно привести к виду

$$Z_0 \approx \frac{60}{V \epsilon_r} \ln(D/d); \quad \beta \approx \pi f \sqrt{\epsilon_r} / 150;$$

$$\alpha \approx (1 + d/D) \sqrt{f} / (24dZ_0) + 1/(96d^2Z_0) + (\beta \operatorname{tg} \delta)/2,$$

где d и D выражены в мм, частота f — в МГц, Z_0 — в Ом, β — в рад/м, α — в Нп/м.

При составлении программы выберем в качестве исходных данных d , D/d , ϵ_r , $(\operatorname{tg} \delta)/2$ и f (при указанной выше нормировке) и разместим их соответственно в регистрах 2, 3, 4, 5 и X, учитывая, что может оказаться необходимым вычисление β и α для нескольких значений частоты. В этом случае первым оператором программы 5 заносим значение частоты из регистра X в регистр 6, а остальную часть программы составляем в соответствии с упрощенными расчетными соотношениями. Результаты вычислений размещаем следующим образом: P7 = β , P8 = Z_0 , PX = α . Значение заданной частоты сохраняется в регистре 6.

Программа 5. Вычисление параметров коаксиальной линии

P6	F3	ln	6	0	\times	\uparrow	F4	$\sqrt{ }$	\div	P8	\uparrow
F2	\times	2	4	\times	$1/x$	\uparrow	F3	\div	$+$	P3	F2
\div	\div	P2	π	\uparrow	F4	$\sqrt{ }$	\times	1	5	0	
\div	\uparrow	F6	\times	P7	\uparrow	F5	\times	\leftarrow	F6	$\sqrt{ }$	\uparrow
F3	\times	\uparrow	F2	$+$	\uparrow	\rightarrow	$+$	C/P	P6	BП	F,

После первого пуска программы (нажатием клавиш В/О и С/П) вычисляются значения Z_0 , β и α . При повторных пусках программы нажатием клавиши С/П для других значений частоты вычисляются только параметры β и α .

Контрольные данные для расчета по программе 5 характеристик коаксиального кабеля (médные проводники, $d = 0,72$ мм; $D = 4,8$ мм; $\epsilon_r = 2,3$; $\operatorname{tg} \delta = 4 \cdot 10^{-4}$) для нескольких значений частоты с указанием времени счета приведены в табл. 44.

Таблица 44

f , МГц	1	10	100	1000
Z_0 , Ом	75,05543	75,05543	75,05543	75,05543
β , рад/м	0,03176307	0,3176307	3,176307	31,76307
α , Нп/м	0,001160763	0,003135207	0,009769884	0,03465994
t , с	12	6	5	6

При решении сложных задач, когда длина программы автоматических вычислений превышает емкость УЗП микрокалькулятора используемого типа, алгоритм решения приходится разбивать на две или более частей, последовательно реализуемых отдельными программами. В этих случаях целесообразно предусматривать такое размещение исходных данных и результатов выполнения каждой программы в запоминающих устройствах, при котором требуется наименьшее число ручных операций перед вводом следующей программы.

Иногда требуется дополнительная обработка результатов вычислений одной из промежуточных программ в обычном режиме. Для этого следует подготовить вывод этих результатов таким образом, чтобы предельно упростить операции обычного режима вычислений, а также затраты времени на анализ пользователем промежуточных результатов.

Пример 50. Составить программу для расчета одиородной линии передачи электромагнитных волн с потерями, нагруженной на произвольное комплексное сопротивление $Z_H = R_H + jX_H$, по известным длине линии и параметрам β и α .

Расчетная формула для входного сопротивления одиородной линии имеет вид

$$Z_{\text{вх}} = Z_0 \frac{(1 - p^2 e^{-4\alpha l}) - 2jpe^{-2\alpha l} \sin(2\beta l - \varphi)}{(1 + p^2 e^{-4\alpha l}) - 2pe^{-2\alpha l} \cos(2\beta l - \varphi)},$$

где коэффициент отражения от нагрузки учитывается значениями его модуля p и аргумента φ :

$$p = \sqrt{\frac{(R_H - Z_0)^2 + X_H^2}{(R_H + Z_0)^2 + X_H^2}}; \quad \varphi = \arctg \frac{2X_H Z_0}{R_H^2 + X_H^2 - Z_0^2}.$$

Отсутствие в используемом входном языке операторов автоматического вычисления обратных тригонометрических функций не позволяет непосредственно включить в программу вычисление φ по заданному сопротивлению нагрузки. Кроме того, емкость УЗП в 60 шагов затрудняет решение задачи с помощью одной программы и его приходится разбивать на два этапа.

Целесообразно ограничить первый этап вычислением модуля и аргумента коэффициента отражения. В этом случае вторая программа будет пригодна для самостоятельного использования при известном значении коэффициента отражения от нагрузки линии.

Для вычисления модуля и аргумента коэффициента отражения необходимо ввести в программу вычисление функции $\arctg \varphi$. Программная память с 60 ячейками не позволяет выполнить все необходимые в этом случае вычисления в рамках одной программы. Поэтому ограничимся в первой программе вычислением модуля p и значения $\operatorname{tg} \varphi = 2X_H Z_0 / (R_H^2 + X_H^2 - Z_0^2)$. Для определения $\varphi = \arctg(\operatorname{tg} \varphi)$ после выполнения этой программы можно обратиться к таблицам тригонометрических функций или вычислить φ по специальной программе для $\arctg x$.

Следует учесть, что в инженерных приложениях основной интерес представляет значение аргумента φ в пределах от 0 до 360° . Поэтому первая из составляемых программ должна подготовить данные, позволяющие по главному значению аргумента, определенному из таблицы, найти искомое значение.

С учетом приведенных замечаний составим программу 6 с размещением исходных данных $Z_0 = P6$, $R_h = P7$, $x_h = P8$. По окончании вычислений значения p и $\operatorname{tg}\phi$ оказываются занесенными в регистры 4 и 7 соответственно, а на индикаторе высвечиваются цифры 7; 180 или 90 в зависимости от знаков числителя и знаменателя выражения, определяющего $\operatorname{tg}\phi$. Если эта дробь положительна, то на индикаторе высвечивается цифра 7, указывающая на номер регистра, в котором находится значение $\operatorname{tg}\phi$, определяемое как главное значение $\phi = \operatorname{arctg}(\operatorname{tg}\phi)$ со своим знаком. Если знаменатель дроби отрицателен, то на индикаторе высвечивается число ± 180 , указывающее, что $\phi = \operatorname{arctg}(\operatorname{tg}\phi) \pm 180^\circ$. При равенстве нулю знаменателя в индикаторе высвечивается 90 ($\phi = \pm 90^\circ$, но знак ϕ определяется знаком мнимой составляющей X_h сопротивления нагрузки).

Программа 6. Вычисление коэффициента отражения

F6	$1/x$	\uparrow	F7	\times	P7	\times	F8	\times	P8	x^2	\uparrow	F7
x^2	$+$	1	$+$	2	\div	P6	\uparrow	F7	$+$	P5	F7	
$-$	\uparrow	F5	\div	$\sqrt{P4}$	F6	1	$-$	$x \neq 0$	P - P2			
\uparrow	F8	XY	\div	P7	F2	$x \geq 0$	FC _x	7	C/P	1	8	
0	C/P	F8	\uparrow	x^2	$\sqrt{\quad}$	\div	9	0	\times	C/P		

После определения (по таблице или с помощью дополнительной программы) главного значения ϕ и его уточнения при необходимости в соответствии с результатами программы 6, в УЗП вводится программа 7, перед пуском которой исходные данные размещают следующим образом: $P2 = 2\beta l$, $P3 = \alpha/\beta$, $P4 = p$, $P5 = \phi$ (в радианах). Если перед этим выполнялась программа 6, то p можно дополнительно не вводить (оно хранится в регистре 4 после выполнения программы 6). После выполнения программы 7 искомые результаты размещают следующим образом: $PX = P6 = |Z_{bx}|/Z_0$, $P7 = R_{bx}/Z_0$, $P8 = X_{bx}/Z_0$, а содержание остальных регистров не изменяется.

Программа 7. Вычисление входного сопротивления однородной линии

F2	\uparrow	F5	$-$	P7	F3	\times	/ - /	e^x	\uparrow	F4	\times
P6	2	\times	\uparrow	F7	sin	\times	/ - /	P8	F7	cos	\times
P7	F6	x^2	1	$+$	P6	2	$-$	T - /	\leftarrow	F6	\uparrow
F7	--	P6	\rightarrow	\uparrow	F6	\div	P7	F8	\uparrow	F6	\div
P8	x^2	\uparrow	F7	x^2	$+$	$\sqrt{\quad}$	P6	C/P	BП	P0	

Предположим, что необходимо вычислить входное сопротивление коаксиального кабеля ($Z_0 = 75 \Omega$, $R_h = 50 \Omega$, $X_h = 30 \Omega$, $\alpha = 0,01 \text{ Нп/м}$, $\beta = 1 \text{ рад/м}$, $l = \pi/10 \text{ м}$). Используя программу 6, получаем на индикаторе 180, после чего вызываем $p = 0,3037835$ (из регистра 4) и $\operatorname{tg}\phi = -2,022471$ (из регистра 7) при времени счета около 10 с. Определив по таблице или другим способом $\phi = 2,0299896$ рад, вводим в УЗП программу 8 и после занесения исходных данных ($P2 = \pi/5$, $P3 = 0,01$, $P5 = 2,0299896$) и выполнения программы (время счета около 12 с) получаем в регистрах 6—8 значения $|Z_{bx}|/Z_0 = 1,097909$; $R_{bx}/Z_0 = 0,9185060$; $X_{bx}/Z_0 = 0,6014585$. Изменяя значения α и β , можно аналогично вычислить входное сопротивление на других частотах.

При использовании микрокалькулятора типа «Электроника Б3-34» с большой емкостью программной памяти и оператором arctg рассматриваемая задача решается с помощью одной программы 8/34 с размещением исходных данных $Z_0 = P0$, $R_h = P1$, $X_h = P2$, $\alpha = P3$, $\beta = P4$, $l = P6$ и результатов вычислений $PX = Z_{bx}$, $PC = R_{bx}$, $PД = X_{bx}$, $P7 = p$, $P8 = \phi$.

Программа 8/34. Вычисление входного сопротивления длинной линии

ИП0	ИП1	-	x^2	ИП2	x^2	+	ВХ	ИП0	ИП1	
+	x^2	+	\div	V	ЗП7	ИП0	ИП2	\times	2	
\times	ИП1	x^2	ИП2	x^2	+	ИП0	x^2	-	\div	
arctg	ЗП8	ИП0	ИП1	-	$x \geq 0$	41	ИП8	π	+	
ЗП8	ИП8	ИП6	ИП4	\times	2	.	\times	-	ЗП9	cos
ИП3	ИП6	\times	2	$/ - /$	\times	e^x	ИП7	\times	ЗПА	
2	\times	\times	ИПА	x^2	$X \rightarrow Y$	-	1	+	ИП0	
\div	ЗПВ	ИП9	sin	ИПА	\times	2	\times	ИПВ	\div	
ЗПД	1	ИПА	x^2	-	ИПВ	\div	ЗПС	x^2	ИПД	
x^2	+	V	С/П	БП	41					

В некоторых случаях число используемых программ удается уменьшить, используя для вычисления различных переменных, связанных некоторым уравнением, одну и ту же программу. Ввод данных в подобные программы следует организовать так, чтобы искомая переменная определялась автоматически.

Для примера рассмотрим уравнение

$$f_p = 1 / (2\pi V LC),$$

связывающее резонансную частоту колебательной системы без потерь с ее параметрами (например, длиной и массой маятника, емкостью и индуктивностью электрического контура и т. п.).

При решении радиотехнических задач обычно выражают частоту f в килогерцах, емкость C — в пикофарадах и индуктивность L — в микрогенри. При таком нормировании переменных уравнение связи принимает вид

$$f_p = 1 / V 4\pi^2 10^{-12} LC.$$

Составляя программу для вычисления одной из переменных, входящей в эту формулу, по известным значениям двух других, коэффициент $4\pi \cdot 10^{-12}$ можно заносить в регистр памяти перед вычислениями (что сократит длину программы, но увеличит время ввода исходных данных) или ввести непосредственно в программу, что уменьшит суммарные затраты времени.

Удобно строить программу таким образом, чтобы она проверяла равенство нулю каждой из введенных в регистры памяти переменных и при выполнении этого условия вычисляла значение этой переменной по отличным от нуля значениям остальных переменных. Подобным образом построена программа 9, перед вводом которой в регистры 2, 3 и 4 соответственно заносят значения L , C и f_p , причем вычисляемой переменной придают значение 0.

Программа 9. Вычисление f , L или C резонансной системы

π	2	ВП	6	$/ - /$	\times	x^2	\uparrow	F4	$x = 0$	$F \div$	F2
\times	\uparrow	F3	\times	$1/x$	V	P4	C/П	БП	P0	x^2	\times
\uparrow	F2	$x = 0$	$/ - /$	F3	\times	$1/x$	P2	C/П	БП	P0	\times
$1/x$	P3	C/П	БП	P0							

После выполнения программы 9 значение искомой переменной заносится в «свой» регистр и высвечивается на индикаторе. Для повторных вычислений по занесенной в УЗП программе 9 с измененными исходными данными нажимают только клавишу С/П.

Решение задач с большим числом исходных данных (например, решение систем уравнений), не вмещающихся в запоминающих устройствах, целесообразно организовать в полуавтоматическом режиме использования микрокалькулятора таким образом, чтобы максимально автоматизировать отдельные этапы вычислений с операциями над такими подмножествами исходных данных или промежуточных результатов, которые могут быть размещены в запоминающих устройствах. Составление программы и организация вычислительного процесса в этих случаях должны обеспечивать минимальные суммарные затраты времени.

В качестве примера использования микрокалькулятора в полуавтоматическом режиме вычислений рассмотрим преобразование симплекс-таблицы (100) на одном шаге модифицированного жорданова исключения, используемого при решении задач линейного программирования, подобных рассмотренным в предыдущих главах. При помощи микрокалькулятора с 13 регистрами памяти можно полностью автоматизировать преобразование симплекс-таблиц, размер которых не превышает 3×3 . В практических задачах размеры симплекс-таблиц обычно больше этого предела и для их преобразования по формулам (133) программируемый микрокалькулятор придется использовать в полуавтоматическом режиме с вычислением только одного элемента новой симплекс-таблицы после каждого пуска программы.

Выбрав разрешающий элемент $\alpha_{pq}^{(k-1)}$ заданной симплекс-таблицы произвольного размера $m \times n$, можно преобразовать ее методом модифицированного жорданова исключения с помощью программы 10. Для первого пуска этой программы после занесения в УЗП нажимают клавиши В/О и С/П, для последующих пусков — только клавишу С/П.

Программа 10. Преобразование симплекс-таблицы по строкам

↑ F6 × ↑ F3 ÷ /—/ ↑ F5 + С/П БП
P0

При использовании программы 10 целесообразно вначале преобразовать разрешающие столбец и строку заданной симплекс-таблицы. Для преобразования разрешающего столбца следует ввести $\alpha_{pq}^{(k-1)} = P_3$, $0 = P_5$, $1 = P_6$, а затем, вводя последовательно в регистр X элементы разрешающего столбца $\alpha_{1q}^{(k-1)}, \dots, \alpha_{mq}^{(k-1)}$ с пуском программы после ввода каждого из этих элементов, записать в новую симплекс-таблицу высвечиваемые значения $\alpha_{1q}^{(k)}, \dots, \alpha_{mq}^{(k)}$ элементов q -го столбца. Для преобразования элементов разрешающей строки следует ввести $-1 = P_6$, а затем, вводя последовательно в регистр X элементы разрешающей строки $\alpha_{p1}^{(k-1)}, \dots, \alpha_{pn}^{(k-1)}$.

с пуском программы после ввода каждого элемента, записать в новую симплекс-таблицу высвечиваемые значения $\alpha_{pq}^{(k)}$, ..., $\alpha_{pn}^{(k)}$ элементов p -й строки.

Элементы остальных (неразрешающих) строк преобразуются следующим образом. Для преобразования элементов i -й строки следует ввести $\alpha_{iq}^{(k-1)} = P_6$, а затем, последовательно вводя пары значений $\alpha_{pj}^{(k-1)} = RX$, $\alpha_{ij}^{(k-1)} = P_5$ с пуском программы после каждого ввода такой пары значений, записать в новую симплекс-таблицу значения элементов $\alpha_{ij}^{(k)}$ преобразуемой строки, которые высвечиваются после каждого выполнения программы.

Процесс преобразования симплекс-таблицы в полуавтоматическом режиме можно ускорить, если вводить в регистры памяти возможно большее число элементов разрешающей строки при последовательном преобразовании симплекс-таблицы по блокам с таким же числом столбцов. Подобная организация вычислений предусмотрена в программе 11 (первый пуск — нажатием клавиш В/О и С/П, последующие — нажатием клавиши С/П), в которой полностью использованы регистры памяти микрокалькулятора.

Программа 11. Преобразование симплекс-таблицы по блокам

P7	F2	$x = 0$	P2	F7	\uparrow	F5	\div	/—/	C/П	БП	P0	
$x < 0$	P ₁		F4	\uparrow	F5	\div	P3	F4	\rightarrow	P4	F3	C/П
БП	P0	F4	\uparrow	F8	\times	\uparrow	F5	\div	/—/	\uparrow	F7	
+	P3	БП	F3									

При работе с этой программой после выбора разрешающего элемента $\alpha_{pq}^{(k-1)}$ заданную симплекс-таблицу произвольного размера $m \times n$ при $n > 8$ разбивают на блоки по семь столбцов (без учета разрешающего столбца), причем последний блок может иметь менее семи столбцов. Записав в вычислительный бланк для преобразованной симплекс-таблицы значение $\alpha_{pq}^{(k)} = 1$, выполняют операции, описываемые следующим алгоритмом:

1. Ввести 0 = P2, $\alpha_{pq}^{(k-1)} = P5$, принять номер строки $i = 1$ и перейти к п. 2.

2. Если $i = p$, то принять $i = i + 1$ и перейти к п. 3.

3. Если $i = m + 1$, то принять номер блока $s = 1$ и перейти к п. 5, иначе к п. 4.

4. Ввести в регистр X элемент разрешающего столбца $\alpha_{iq}^{(k-1)}$, пустить программу, записать в вычислительный бланк высвечиваемое значение RX = $\alpha_{iq}^{(k)}$, принять $i = i + 1$ и перейти к п. 2.

5. Ввести в регистр 2 любое отрицательное число, ввести в стек по часовой стрелке элементы разрешающей строки s -го блока (если число столбцов блока меньше семи, то довернуть стек по часовой стрелке так, чтобы вывести элемент $\alpha_{p1}^{(k-1)}$ в регистр X), заслать высвеченное значение $\alpha_{p1}^{(k-1)}$ в регистр 4, принять номер столбца в блоке $j = 1$ и перейти к п. 6.

6. Пустить программу, записать в бланк значение РХ, $\alpha_{pl}^{(k)}$, принять $j = j + 1$ и перейти к п. 7.

7. Если j больше числа столбцов в s -м блоке, то перейти к п. 8, иначе к п. 6.

8. Ввести в регистр 2 любое положительное число, принять $i = 1$ и перейти к п. 9.

9. Если $i = p$, то принять $i = i + 1$ и перейти к п. 10.

10. Если $i = m + 1$, то принять номер блока $s = s + 1$ и перейти к п. 5, иначе ввести $\alpha_{iq}^{k-1} = P8$, принять $j = 1$ и перейти к п. 11.

11. Ввести в регистр X значение α_{ij}^{k-1} , пустить программу, записать в бланк РХ = $\alpha_{ij}^{(k)}$, принять $j = j + 1$ и перейти к п. 12.

12. Если j больше числа столбцов в s -м блоке, то принять $i = i + 1$ и перейти к п. 9, иначе к п. 11.

В связи с меньшим числом требуемых вводов суммарные затраты времени на преобразование симплекс-таблицы по программе 11 минимальны. Подобную организацию вычислений с предельной автоматизацией вспомогательных ручных операций целесообразно использовать и при решении других задач с большим числом исходных данных.

Особое значение имеет оптимизация алгоритма принятия решения для эвристических программ, содержащих большое число условных и безусловных переходов. Примером может служить алгоритм игры в «крестики — нолики» (см. рис. 25), реализуемый программой 12

Программа 12. Игра в «крестики — нолики»

9	↑	C/P	PП	F8	P2	C/P	ПП	/—/	$x < 0$	F2	F2
P7		ПП	P8	C/P	ПП	/—/	$x < 0$	4	ПП	P8	C/P
/—/	0	C/P	F8	5	ПП	P—	F8	7	7	C/P	P7
F8	4	ПП	P—	F7	—	$x = 0$	F5	π	×	cos	B/O
F7	1	—	$x \neq 0$	F9	$x < 0$	9	8	+	↑	P8	B/O

Основной вычислительный блок алгоритма реализован в программе 12 подпрограммой:

$$| 1 - x \neq 0 \ A_1 \ x < 0 \ \overbrace{A_2 \ 8 + \uparrow}^{\uparrow} \ P8 \ B/O |.$$

Для вычисления $y = [x - a]_8$ при $a \neq 1$ эта подпрограмма используется со второго оператора с предварительным вызовом требуемого значения a в регистр X.

Самостоятельное значение имеет использованный в рассматриваемой программе блок проверки четности числа, хранимого в регистре X, который содержит всего шесть операторов: $|\uparrow \pi \times \cos x < 0 B|$, где B — оператор передачи управления при нечетности анализируемого числа.

После ввода программы 12 в УЗП следует нажать клавиши B/O и C/P, что приведет к высвечиванию номера $y_1 = 9$ клетки, занимаемой первым ходом микрокалькулятора (см. рис. 24). Играющий

набирает номер занимаемой им клетки и нажимает клавишу С/П, повторяя эти операции после каждого высвечивания ответа микрокалькулятора. Игра продолжается либо до выигрыша микрокалькулятора, индицируемого высвечиванием числа 77 (в этом случае последний ход микрокалькулятора хранится в регистре Y и может быть вызван нажатием клавиши $X \leftrightarrow Y$), либо до ничейного результата, при котором на индикаторе высвечивается 0.

При рациональном использовании входного языка в УЗП микрокалькулятора типа «Электроника Б3-21» (и, тем более, типа «Электроника Б3-34») можно разместить программу игры «Ним» [5], хорошо известную специалистам по программированию ЭВМ и часто используемую для оценки эвристических возможностей машинных языков.

5. ТИПОВЫЕ АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ

Основой организации вычислительного процесса в цифровых ЭВМ являются простые и условные (содержащие некоторое условие) арифметические выражения, которые описывают на входном языке ЭВМ действия над исходными данными, требуемые для вычисления новых значений текущих переменных. Программа в целом, как и имеющие смысл во входном языке ее фрагменты, являются арифметическими выражениями. Среди бесконечного числа арифметических выражений можно выделить типовые, обеспечивающие реализацию наиболее часто встречающихся вычислительных процедур. Такие выражения могут представлять собой как самостоятельные программы, так и фрагменты или подпрограммы более сложных программ. Умение составлять типовые арифметические выражения существенно упрощает составление сложных программ автоматических вычислений.

Внутренние микропрограммы, реализующие операторы вычисления тригонометрических функций, иногда предусматривают ввод аргумента только в радианах или, реже, только в градусах. Поэтому возникает необходимость перевода радиан в градусы или наоборот. Перевод смешанного десятичного представления угла в градусах в радианы реализуется арифметическим выражением $| \uparrow |L| \pi | \cdot | \times |180 | \div |$. Если угол задан целыми числами α° градусов и α' минут и, в общем случае, смешанным числом α'' секунд, то перевод в радианы согласно второй из формул (86а) реализуется арифметическим выражением

$$| \uparrow |6|0| \div | \uparrow |F3| + |6|0| \div | \uparrow |F2| + | \uparrow |\pi | \times |1|8|0| \div |$$

с исходными данными $\alpha^\circ = P2$, $\alpha' = P3$ и вводом числа α'' непосредственно в арифметическое выражение (если последнее оформлено в виде программы, то α'' вводят непосредственно в индикаторный регистр).

По этому арифметическому выражению, например, для $\alpha = 57^\circ 17' 45''$ результат $\rho = 1$ в радианах вычисляется микрокаль-

кулятором примерно за 4 с, для $\alpha = 57^\circ 17'$ время счета ($\rho = 0,9997828$) около 5 с, а для $\alpha = 57^\circ$ результат $\rho = 0,9948376$ вычисляется примерно за 6 с. Это странное на первый взгляд соотношение затрат времени объясняется тем, что операции над нулевыми значениями операндов ($\alpha' = 0, \alpha'' = 0$) в микрокалькуляторе выполняются медленнее, чем над отличными от нуля числами.

Перевод дуговой меры в радианах в угловую в градусах реализуется арифметическим выражением $| \uparrow | \pi | \div | 1 | 8 | 0 | \times |$. Однако при необходимости вычисления целых чисел градусов и минут задача усложняется, так как необходимо иметь выражение, выделяющее целую часть смешанного десятичного представления числа. Для этого можно воспользоваться округлением результата сложения смешанного числа x с вспомогательным 8-значным числом. Если выполняется условие $a + x < 10^9$, то полученное 8-значное число равно $a + E(x)$ и вычитание из него a дает в результате искомую целую часть $E(x)$. Описанный алгоритм реализуется фрагментом $\uparrow | 1 | \text{ВП} | 7 | \text{XY} | + | \text{XY} | - |$, где $a = 10^{7*}$. Используя подобный фрагмент, нетрудно составить программу 13.

Программа 13. Перевод радиан в градусы, минуты и секунды

\uparrow	π	\div	1	8	0	\times	ПП	$F \times$	ПП	$F \times$	\rightarrow
\rightarrow	C/P	BП	P0	P2	1	BП	7	XY	+	XY	-
\uparrow	\leftarrow	F2	XY	-	6	0	\times	B/O			

Для этой программы исходное число радиан заносят в индикаторный регистр, а результат заносится программой в регистры $PX = \alpha^\circ$, $C1 = \alpha'$, $C2 = \alpha''$. Для вызова на индикатор α' и α'' достаточно дважды ввести оператор \leftarrow — поворота стека против часовой стрелки. Программа вычисляет α (1 рад) = $57^\circ 17' 44,8008$ за 10 с.

Для микрокалькуляторов «Электроника Б3-21» первых выпусков в программе 13 следует заменить блок вычисления $E(x)$, записанный по адресам 25 — 40 фрагментом, приведенным в сноске.

В программе 13а на индикатор выводится полный результат вычислений — целая часть высвечиваемого числа равна α° , минуты указаны в первых двух разрядах после запятой, а секунды — в следующих разрядах.

Программа 13а. Перевод радиан в градусы, минуты и секунды с выводом результата на индикатор

\uparrow	π	\div	1	8	0	\times	ПП	P,	ПП	P,	ПП
P3	PП	P3	C/P	BП	P0	1	0	0	\div		
+	B/O	P2	1	BП	7	XY	+	XY	-	\uparrow	\rightarrow
F2	XY	-	6	0	\times	B/O				\uparrow	\leftarrow

* В первых выпусках микрокалькуляторов типа «Электроника Б3-21» в связи с неконтролируемым переполнением выделение целой части приведенного фрагмента сопровождается округлением результата. Поэтому его следует заменить для такого микрокалькулятора фрагментом $| P2 | 1 | \text{ВП} | 7 | \text{XY} | - | \uparrow | F2 | \text{XY} | P2 |$ — $| x < 0 | P3 | F2 | 1 | - | P2 | F2 | \text{C/P} |$.

Программа 13а вычисляет $\alpha(1) = 57,17448$ ($\alpha = 57^\circ 17' 44,8''$) за 11 с.
Для микрокалькуляторов первых выпусков подпрограмму выделения целой части с адреса 52 следует заменить фрагментом

$$\begin{array}{c} | - | - | \uparrow | F2 | X Y | P3 | - | x < 0 | PC_x | F3 | 1 | \\ | - | P3 | F3 | \uparrow | \leftarrow | F2 | X Y | - | 6 | 0 | \times | B/O | . \end{array}$$

Многие инженерные задачи связаны с анализом функциональных зависимостей, описанных степенными алгебраическими многочленами. Численный анализ в этом случае сводится к вычислению многочлена при различных значениях аргумента. Для этого целесообразно использовать формулу (33), реализуемую (после занесения аргумента в индикаторный регистр и коэффициентов многочлена в стек) арифметическим выражением $\uparrow F8 \times \uparrow F7 \leftarrow P7 +$, где регистр 7 использован в качестве буферного. Если коэффициенты многочлена заносятся только в ЗУПВ, то в подпрограмму целесообразно выделить выражение вида $+ \uparrow F8 \times \uparrow$, выполняющее сложение заносимых в регистры X и Y величин $y_k x$ и a_i и умножение их суммы на значение аргумента x .

Оба приведенных выражения можно объединить в одну подпрограмму так, что в первом из них содержится обращение ко второму: $| ПП A F7 \leftarrow P7 + B/O + \uparrow F8 \times \uparrow B/O$. В этом выражении символом A обозначен оператор передачи управления первому слева оператору \uparrow , а входами в подпрограмму являются оператор ПП, второй слева оператор $+$ и первый слева оператор \uparrow . С помощью такого выражения вычисление алгебраических многочленов до степени 10 включительно реализуется универсальной программой 14 с исходными данными $a_{10} = C1, a_9 = C2, a_8 = C3, a_7 = C4, a_6 = C5, a_5 = C6, a_4 = P2, a_3 = P3, a_2 = P4, a_1 = P5, a_0 = P6, x = PX$.

Программа 14. Вычисление многочленов степени $m \leq 10$

P8	\leftarrow	P7	ПП	$/-$	ПП	$/-$	ПП	$/-$	ПП	$/-$	ПП
$/-$	\uparrow	F7	\leftarrow	ПП	FC _x	F2	ПП	PC _x	F3	ПП	PC _x
F4	ПП	PC _x	F5	ПП	PC _x	F6	+	C/P	БП	P0	\uparrow
F8	\times	\uparrow	F7	\leftarrow	P7	$+$	B/O	$+$	\uparrow	F8	\times
\uparrow	B/O										

Программа 14 автоматически заносит значение аргумента в регистр 8, а регистр 7 используется как буферный для перезаписи содержимого стека. Значение многочлена $y(3) = 2x^{10} + 3x^9 + 4x^8 + 5x^7 + 6x^6 + 7x^5 + 8x^4 + 9x^3 + 10x^2 + 11x + 12 = 221427$ программа вычисляет за 15 с.

При заданном шаге изменения аргумента Δx программу 14 можно дополнить фрагментом $F8 \uparrow \Delta x +$, записываемым в конце программы между операторами С/П и БП*. В этом случае для повторного вычисления многочлена при $x_{i+1} = x_i + \Delta x$ достаточно нажать клавишу С/П. Если некоторые коэффициенты многочлена сте-

* При соответствующем изменении адресов обращения к подпрограмме.

пени 10 равны нулю, то для сокращения времени вычислений можно составлять упрощенные варианты программы (подобные, например, программе 3).

Нормирование многочлена степени 14 делением на коэффициент при x^{11} позволяет составить программу 15 для вычисления значения нормированного многочлена $x^{11} + a'_1 x^{10} + a'_9 x^9 + a'_8 x^8 + a'_7 x^7 + a'_6 x^6 + a'_5 x^5 + a'_4 x^4 + a'_3 x^3 + a'_2 x^2 + a'_1 x + a'_0$.

Программа 15. Вычисление нормированного многочлена степени 11

Для вычисления значений алгебраических многочленов до степени 11 включительно на микрокалькуляторе типа «Электроника Б3-34» можно воспользоваться программой для многочлена пятой степени, приведенной в примере 48. Следует лишь заменить число 6, записываемое в регистр O оператором по адресу 01, числом 12 и адрес перехода 05 после оператора $x = 0$ на адрес 06. Исходные данные $a_0 = P1$, $a_1 = P2$, $a_2 = P3$, $a_3 = P4$, $a_4 = P5$, $a_5 = P6$, $a_6 = P7$, $a_7 = P8$, $a_8 = P9$, $a_9 = PA$, $a_{10} = PB$, $a_{11} = PC$. Регистр O используется для хранения текущего адреса косвенных обращений к регистрам ЗУПВ, а регистр D — для записи заданного значения аргумента, вводимого перед пуском программы в индикаторный регистр.

Описанные выше арифметические выражения можно использовать и для вычисления дробно рациональной функции

$$y = \frac{a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0}{b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0},$$

часть коэффициентов которой могут быть равными нулю, с помощью программы 16 с исходными данными $a_4 = P2$, $a_3 = P3$, $a_2 = P4$, $a_1 = P5$, $a_0 = P6$, $b_5 = C1$, $b_4 = C2$, $b_3 = C3$, $b_2 = C4$, $b_1 = C5$, $b_0 = C6$, $x = PY$. Результат вычислений выводится на индикатор.

Программа 16. Вычисление дробно-рациональной функции

При необходимости можно составить аналогичную программу для вычисления дробно-рациональных функций с большей степенью числителя и меньшей знаменателя или наоборот. Нормирование функции сокращает число отличных от единицы коэффициентов и позво-

ляет повысить степени нормированных многочленов в числителе и знаменателе.

Программы 13—16 целесообразно использовать для вычислений при ряде значений аргумента. Если требуется однократное или, во всяком случае, немногократное повторение вычислений, то можно снять ограничения на предельные степени многочленов. Для этого достаточно накапливать промежуточный результат вычислений согласно формуле (33) по мере введения коэффициентов в память микрокалькулятора. При этом число ручных операций ввода коэффициента не увеличивается. Подобный прием использован в программе 17 с исходными данными $m = P3$, $a_0 = P8$, $x = P2$ (в качестве вспомогательных используются регистры 4, 6 и 7).

Программа 17. Вычисление многочлена произвольной степени

P6	0	P7	F3	P4	F6	↑	F7	+	↑	F2	×
P7	F4	1	—	P4	$x = 0$	\div	F7	↑	F8	+	C/P
БП	P1										

После занесения в УЗП программы 17 в регистр X последовательно вводят коэффициент a_m с пуском программы клавишами В/О и С/П и коэффициенты a_{m-1}, \dots, a_1 с пуском программы после каждого ввода нажатием только клавиши С/П. После каждого выполнения программы на индикаторе высвечивается степень при вводимом следующим коэффициенте. После высвечивания цифры 1 вводят коэффициент a_1 и после выполнения программы высвечивается искомое значение многочлена степени m . Значения m , a_0 и x сохраняются в регистрах памяти и для повторного вычисления (например, при другом значении x) повторяют ввод коэффициентов с нажатием после ввода a_m клавишей В/О и С/П.

Аналогичный способ используется и в программе 18 для вычисления дробно-рациональной функции с произвольными степенями m числителя и n знаменателя.

Программа 18. Вычисление дробно-рациональной функции произвольной степени

→	F7	ПП	P ÷	F8	+	→	F4	C/P	→	F4	ПП
P ÷	F5	+	↑	←	X Y	÷	C/P	P3	0	↑	←
+	↑	F2	×	→	F3	1	—	P3	$x \neq 0$	PBP	C/P
БП	F ÷	←	↑	B/O							

Перед пуском программы 18 вводят степени $m = P7$ числителя и $n = P4$ знаменателя, их свободные члены $a_0 = P8$, $b_0 = P5$ и значение аргумента $x = P2$. После ввода исходных данных последовательно вводят коэффициент a_m с пуском программы клавишами В/О и С/П и коэффициенты $a_{m-1}, \dots, a_1, b_n, \dots, b_1$ с пуском программы (только клавишей С/П) после каждого ввода. После каждого промежуточного ввода на индикаторе высвечивается степень при вводимом следующем коэффициенте. После ввода коэффициента b_1

и выполнения программы на индикаторе высвечивается искомое значение функции. Например, для вычисления функции

$$y(0,5) = \frac{x^4 + 5x^2 + 4x + 8}{2x^3 + 3x^2 + x + 3} = 2,513888$$

необходимо выполнить операции, описываемые следующим предложением (в скобках указаны высвечиваемые числа): 0,5 Р2 4 Р7 3 Р4 8 Р8 3 Р5 1 В/О С/П (3) 0 С/П (2) 5 С/П (1) 4 С/П (3) 2 С/П (2) 3 С/П (1) 1 С/П (2,513888).

При операциях с комплексными числами приходится переходить от алгебраической формы их представления $a + jb$ к комплексной форме $\sqrt{a^2 + b^2} e^{j\varphi}$. При этом во многих практических приложениях аргумент $\varphi = \operatorname{arctg} b/a$ при $a > 0$ или $\varphi = \pi - \operatorname{arctg} b/a$ при $a < 0$ должен быть определен в интервале 2π . Если при этом берется главное значение $\operatorname{arctg} b/a$, лежащее в пределах $]-\pi/2, \pi/2[$, то интервал определения аргумента $]-\pi/2, 3\pi/2[$. В универсальных программах следует также предусматривать случай $a = 0$, соответствующий $\varphi = \pi/2$ при $b > 0$ и $\varphi = -\pi/2$ при $b < 0$. Определение аргумента на интервале $]-\pi, \pi[$ требует анализа как вещественной, так и мнимой частей комплексного числа: $\varphi = \operatorname{arctg} b/a$ при $a > 0$, $\varphi = \pi + \operatorname{arctg} b/a$ при $a < 0$ и $b > 0$ и $\varphi = -\pi + \operatorname{arctg} b/a$ при $a < 0$, $b < 0$.

Отсутствие в используемом входном языке операторов вычисления обратных тригонометрических функций заставляет в простейшем случае ограничиваться вычислением только функции $b/a = \operatorname{tg} \varphi$ с определением формулы вычисления аргумента в интервале 2π . Универсальная программа 19 определяет модуль заданного комплексного числа $a + jb$, отношение $b/a = \operatorname{tg} \varphi$ и указывает формулу вычисления аргумента φ в интервале $]-\pi, \pi[$ при любых значениях и знаках a и b , вводимых соответственно в регистры 2 и 3 перед пуском программы.

Программа 19. Преобразование комплексного числа

F2	x^2	\uparrow	F3	x^2	$+$	\sqrt	P4	9	0	P5	F3
\uparrow	$x < 0$		P3	F5	$/-$	P5	F2	$x \neq 0$	P/-	\div	P6 F2

$x \geqslant 0$ F, 0 P5 F5 2 \times P5 F5 C/П БП Р0

После выполнения программы 19 модуль хранится в регистре 4, а $\operatorname{tg} \varphi = b/a$ в регистре 6. На индикатор выводится 0, если $\varphi = \operatorname{arctg} b/a$, $\pm 90^\circ$, если $\varphi = \pm 90^\circ$, или $\pm 180^\circ$, если $\varphi = \pm 180^\circ + \operatorname{arctg} b/a$. При $a = -3$; $b = -4$ программа вычисляет $\sqrt{a^2 + b^2} = 5$; $b/a = 1,333333$; $\varphi = -180^\circ$ (что соответствует $\varphi = -180^\circ + \operatorname{arctg} 1,333333$) за 6 с.

Программу 19 можно совместить с одной из программ приближенного вычисления $\operatorname{arctg} x$. В частности, комбинированная программа 20 выводит на индикатор значение модуля и заносит в регистр 4 значение выраженного в градусах аргумента комплексного числа, определенное на интервале $]-\pi, \pi[$.

Программа 20. Вычисление модуля и аргумента комплексного числа

F2	$x < 0$	\uparrow	1	8	0	P5	F3	X Y	\div	P4	$x \geq 0$
F \times	F5	$/-$	P5	F4	x^2	0	,	8	0	5	\times
1	$+$	P6	F4	x^2	1	$+$	$\sqrt{}$	\uparrow	F6	$+$	$\sqrt{}$
\uparrow	F4	X Y	\div	8	0	,	6	7	\times	\uparrow	F5
$+$	P4	F2	x^2	\uparrow	F3	x^2	$+$	$\sqrt{}$	P5	C/P	

Перед пуском программы вещественная a и мнимая b части преобразуемого комплексного числа заносятся соответственно в регистры 2 и 3. В программе не предусмотрено преобразование мнимого числа ($a = 0$), поэтому при его преобразовании происходит переполнение. Контрольные результаты выполнения программы 20 для ряда комплексных чисел и погрешности вычисления аргумента приведены в табл. 45 ($M = P_X = P_5$, $\varphi = P_4$).

Последовательное умножение комплексных чисел, представленных в алгебраической форме, выполняется по программе 21.

Таблица 45

a	b	M	φ°	$t, \text{ с}$	$\varphi_{\text{точн}}^\circ$	$\delta\varphi^\circ$
1	1	1,414213	44,96112	10	45	-0,03888
-1	1	1,414213	135,0388	10	135	0,03888
-1	-1	1,414213	-135,0388	10	-135	-0,03888
1	-1	1,414213	-44,96112	10	-45	0,03888
$\sqrt{3}$	1	1,999999	29,92069	11	30	-0,07931
$-\sqrt{3}$	1	1,999999	150,0793	10	150	0,07931
1	$-\sqrt{3}$	1,999999	-60,04453	10	-60	-0,04453
2	0	2	0	20	0	0
-2	0	2	-180	19	-180	0
-2	50	50,03998	92,32723	10	92,290605	0,036625
50	2	50,03998	2,280502	10	2,290605	-0,010103
-2	-50	50,03998	-92,32723	10	-92,290605	-0,036625

Программа 21. Умножение комплексных чисел

F3	\uparrow	F5	\times	\leftarrow	F4	\times	\leftarrow	F2	\uparrow	F5	\times
\uparrow	\rightarrow	$+$	P5	F2	\uparrow	F4	\times	\uparrow	\rightarrow	-	P4

Для вычисления $(x + iy) = (a + jb)(c + jd)$ по этой программе исходные данные $a = P_2$, $b = P_3$, $c = P_4$, $d = P_5$. Вещественная часть результата засыпается программой в регистры X и 4, мнимая — в регистр 5. Время одного умножения около 5 с. Для умножения результата на следующий сомножитель достаточно записать последний в регистры 2 и 3 и повторить выполнение программы. Например, вычисление произведения $(5 + j3)(6 + j8)(3 - j5)(2 + j0)(0 + j4) = -1152 + j2464$ с помощью программы 21 реализу-

ется вводом предложения: 5 P2 3 P3 6 P4 8 P5 В/О С/П (308) F6 (144) 2 P2 0 P3 С/П (616) F6 (288) 0 P2 4 P3 С/П (-1152) F6 (2464).

Деление двух комплексных чисел, представленных в алгебраической форме $(a + jb)/(c + jd) = x + jy$, реализуется программой 22 при $a = P7$, $b = P8$, $c = P4$, $d = P5$.

Программа 22. Деление комплексных чисел

F4	x^2	↑	F5	x^2	+	P6	F7	↑	F5	×	←
F4	×	←	F8	↑	F5	×	↑	→	+	↑	F6
÷	P7	F8	↑	F4	×	↑	→	-	↑	F6	÷
P8	F7	C/П	БП	P0							

После выполнения программы 22 содержание регистров $P7 = x$, $P8 = y$, $P6 = c^2 + d^2$, $P4 = c$, $P5 = d$. Это позволяет выполнять многократное деление комплексных чисел, для чего после деления достаточно вписать новый делитель в регистры 4 и 5 соответственно и нажать клавишу С/П. Например, операции вычисления $(5 - j3)/(5 + j3)/(2 - j4)/(0 + j0,2) = 0,02941177 - j1,117647$ с помощью программы 22 описываются предложением: 5 P7 3 /-/ P8 5 P4 3P5 В/О С/П (0,4705882) F8 (-0,8823529) 2 P4 4 /-/ P5 С/П (0,2235294) F8 (0,005882355) С/П 0 P4 4 /-/ P5 С/П (0,02941177) F8 (-1,117647). Программа выполняет деления соответственно за 7,8 и 13 с.

Деление комплексного числа можно заменить умножением делимого на число, обратное делителю, что позволяет использовать в программе 21 условный оператор для различения кода операции деления или умножений. Подобный прием реализован в программе 23, перед пуском которой в регистры 2 и 3 вводят соответственно вещественную и мнимую части первого числа, а в регистры 4 и 5 — второго. Кроме того, в регистр X вводится код операции (1 для умножения и 2 для деления), определяющий действия над этими числами.

Программа 23. Умножение и деление комплексных чисел

↑	1	—	$x \neq 0$	÷	F2	x^2	↑	F3	x^2	+	P8
F2	↑	F8	÷	P2	F3	/-	↑	F8	÷	P3	F3
↑	F5	×	←	F4	×	←	F2	↑	F5	×	↑
→	+	P5	F2	↑	F4	×	↑	→	-	P4	C/П
БП	P0										

Результат заносится программой 23 в регистры 4 и 5 и после введения в регистры 2 и 3 нового числа и в регистр X — кода операции вычисления можно продолжить. Умножение выполняется за 5 с., деление за 8—9 с. Например, вычисление $(5 - j3)(3 + j2)/(5 + j3)(2 - j4) = 0,6588235 + j0,4647058$ выполняется примерно за 30—40 с.

Умножение и деление комплексных чисел в показательной форме реализуется программой 24. Модуль и аргумент первого числа пе-

перед пуском программы заносят соответственно в регистры 2 и 3, второго числа — в регистры 4 и 5, а в индикаторный регистр — код операции (1 для умножения и 2 для деления).

Программа 24. Умножение и деление комплексных чисел в показательной форме

↑	1	$x=0$	P3	F2	↑	F4	\times	P2	F3	↑	
F5	+	P3	C/P	БП	P0	F2	↑	F4	÷	P2	F3
↑	F5	—	P3	C/P	БП	P0					

Результат заносится программой в регистры 2 и 3, что позволяет ввести следующее число в регистры 4 и 5 и продолжить вычисления.

В инженерной практике иногда возникает необходимость вычисления многочлена с вещественными коэффициентами при комплексном значении аргумента $z = x + jy$. В этом случае целесообразно использовать программу 25, вычисляющую значения многочленов с комплексным аргументом до степени 6 включительно, причем аргумент, в частности может быть вещественным или мнимым числом. Перед пуском этой программы размещение исходных данных $a_6 = P8$, $a_5 = C1$, $a_4 = C2$, $a_3 = C3$, $a_2 = C4$, $a_1 = C5$, $a_0 = C6$, $x = P2$, $y = P3$.

Программа 25. Вычисление многочлена степени 6 комплексного аргумента

F8	↑	F3	\times	P5	F2	\times	↑	F8	\leftarrow	P8	+
P4	ΠΠ	,	ΠΠ	,	ΠΠ	,	ΠΠ	,	ΠΠ	;	C/P
F8	\leftarrow	P8	БП	P0	F3	↑	F4	\times	P6	F5	\times
P7	F2	↑	F5	↑	F6	$+$	P5	F2	↑	F4	
\times	↑	F7	—	↑	F8	\leftarrow	P8	+	P4	B/O	

Вещественная часть результата выводится программой в индикаторный регистр и регистр 4, мнимая — в регистр 5. Программа 25 вычисляет $f(1 + j) = z^6 + 2z^5 + 3z^4 + 4z^3 + 5z^2 + 6z + 7 = -15 + j$ примерно за 30 с. Если степень $m < 6$, то программу 25 следует изменить так, чтобы до обращения к подпрограмме в регистры 4 и 5 были занесены произведения коэффициента a_m при высшей степени на x и y соответственно.

Для вычисления многочлена произвольной степени m с вещественными коэффициентами и комплексным аргументом предназначена программа 26 с исходными данными $x = P2$, $y = P3$, $m = P7$, $a_0 = P8$.

Программа 26. Вычисление многочлена степени m с комплексным аргументом

↑	F3	\times	→	F2	\times	→	F7	1	—	P6	C/P
↑	\leftarrow	+	↑	F3	\times	P4	F2	\times	←	↑	F3
×	P5	F2	\times	↑	F4	+	→	↑	F5	—	→
F6	1	—	P6	$x=0$	F8	\leftarrow	↑	F8	+	↑	←
XY	C/P	БП	P2								

Порядок действий при вычислениях этой программы и программы 17 аналогичны. После выполнения программы вещественная и мнимая части результата хранятся в регистрах X и Y соответственно. Исходные значения x, y, m и a_0 сохраняются в своих регистрах. Несколько иной порядок работы при использовании программы 27.

Программа 27. Вычисление дробно-рациональной функции произвольного порядка с комплексным аргументом

P4	F6	P5	\leftarrow	\uparrow	F3	\times	P7	F2	\times	\leftarrow	\uparrow
F3	\times	P8	F2	\times	\uparrow	F7	$+$	\rightarrow	\uparrow	F8	$-$
\uparrow	F4	$+$	\rightarrow	F5	1	$-$	P5	$x \neq 0$	6	F4	C/P
P4	БП	0	\leftarrow	P2	x^2	\uparrow	\leftarrow	P3	x^2	$+$	1/x
\uparrow	F2	\times	P2	F3	$/-$	/	P3	0	БП	P0	

После записи составляющих $x = P2$, $y = P3$ комплексного аргумента и суммы степеней $m + n = P6$ числителя и знаменателя необходимо занести $0 = C2$, $a_m = C1$, $a_{m-1} = RX$ и, нажав клавиши В/О и С/П, дождаться выполнения первой части программы с высвечиванием на индикаторе коэффициента a_{m-1} . Повторяя ввод в регистр X коэффициентов $a_{m-2}, a_{m-3}, \dots, a_0$ с нажатием клавиши С/П после каждого ввода, следует закончить вычисление числителя (на индикаторе высветится значение a_0).

Затем следует ввести $0 = C2$, $b_n = C1$, $b_{n-1} = RX$ и после нажатия клавиши С/П — коэффициенты b_{n-2}, \dots, b_0 с нажатием клавиши С/П после каждого ввода. После ввода b_0 и выполнения программы искомый результат хранится в стеке: $C1 = \operatorname{Re} f(z)$, $C2 = \operatorname{Im} f(z)$.

После окончания вычислений в регистре 6 сохраняется значение $m + n$, а в регистрах 2 и 3 — вещественная и мнимая части величины, обратной знаменателю. Правильность введения и использования программы 27 можно проверить по результату вычисления $f(1 + j) = (z^8 + 4z^2 + 8z + 15)/(2z^8 + 3z^2 + 4z + 5) = 1,615384 - j0,9230769$.

Вычисление цепных дробей общего вида (36) с числовыми коэффициентами проще всего организовать с помощью программы

$$\uparrow \text{ С/П } XY \div \uparrow \text{ С/П } + \text{ БП } P0.$$

После занесения этой программы, реализующей схему Горнера для деления по формуле (116а), в индикаторный регистр последовательно вводят элементы цепной дроби $\alpha_m, \beta_m, \alpha_{m-1}, \beta_{m-1}, \dots, \alpha_1, \beta_1, \alpha_0$ с нажатием клавишей В/О и С/П после ввода α_m и только С/П после ввода остальных элементов. После выполнения всех операций на индикаторе высвечивается искомое значение цепной дроби. Напри-

мер, после занесения в УЗП этой программы вычисление цепной дроби

$$y = \left[5; \frac{8}{4}, \frac{2}{7}, \frac{3}{5} \right]$$

реализуется предложением: 5 В/О С/П 3 С/П 7 С/П 2 С/П 4 С/П 8 С/П 5 С/П (6,876543).

По рассмотренной программе можно вычислять цепные дроби при заданном числе m звеньев. Несмотря на простоту этой программы, дальнейшая автоматизация вычисления таких цепных дробей нецелесообразна (кроме возможного введения счетчика числа звеньев или элементов), так как не приводит к уменьшению ручных операций при вводе исходных данных.

Число m звеньев цепной дроби, вычисляемой автоматически, ограничено емкостью запоминающих устройств. Исключение составляют лишь цепные дроби некоторых частных видов. Периодическая часть цепной дроби вида

$$y = [\alpha_0; (\beta/\alpha)]$$

с числовыми элементами вычисляется с помощью арифметического выражения 0 P8 F3 \uparrow F8 + \uparrow F2 X \rightleftharpoons Y \div \uparrow F8 X \rightleftharpoons Y P8 — $x = 0 A F8$, где оператор A передает управление третьему слева оператору F3, а коэффициенты β и α предварительно занесены в регистры 2 и 3 соответственно. Дополнив это выражение фрагментом, складывающим периодическую часть дроби с элементом α_0 , получим программу 28, обеспечивающую точность вычислений до семи верных цифр. Исходные данные $\beta = P2$, $\alpha = P3$, $\alpha_0 = P4$, результат выводится на индикатор.

Программа 28. Вычисление бесконечной цепной дроби

0 P8 F3 \uparrow F8 + \uparrow F2 XY \div \uparrow F8
XY P8 — $x = 0$ P \uparrow F8 \uparrow F4 + C/П \uparrow F8

Цепную дробь $[4; (5/3)] = 5,192582$ программа 28 вычисляет примерно за 38 с.

В практических расчетах приходится использовать функциональные цепные дроби, элементы которых являются функциями аргумента x и номера звена k . По программе 29 можно вычислять цепные дроби вида

$$y(x) = \left[\alpha_0; \frac{b_1 x}{\alpha_1}, \dots, \frac{b_m x}{\alpha_m} \right]$$

с числом звеньев $m \leq 5$ ($\alpha_0 = C1$, $\alpha_1 = C2$, $\alpha_2 = C3$, $\alpha_3 = C4$, $\alpha_4 = C5$, $\alpha_5 = C6$, $b_1 = P2$, $b_2 = P3$, $b_3 = P4$, $b_4 = P5$, $b_5 = P6$, $x = PX$).

Программа 29. Вычисление цепной дроби с элементами $\beta_k = b_kx$

P7 0 ↑ ПП Р, F6 ПП Р, F5 ПП Р, F4
 ПП Р, F3 ПП Р, F2 ПП Р, F8 → XY C/Π
 БП Р0 $x \neq 0$ F—/ XY ÷ ↑ F7 × ↑ F8 →
 P8 + ↑ B/O

Если $m < 5$, то вместо недостающих элементов в соответствующие регистры памяти заносят нули. Результат выводится на индикатор, а при повторных пусках программы для других значений x нажимают только клавишу С/Π. При $x = 1$ программа 29 пригодна для вычислений цепных дробей с числовыми элементами. Например, цепная дробь

$$\left[0, \frac{1}{1}, \frac{1}{3}, \frac{4}{5}, \frac{9}{7}, \frac{16}{9}\right] = 0,7855856$$

вычисляется по этой программе примерно за 16 с.

В некоторых случаях числовые элементы цепной дроби целесообразно записать непосредственно в программу для увеличения числа звеньев и упрощения ввода исходных данных. В качестве примера приведем программу 30, вычисляющую функцию $\arctg x$ по цепной дроби (52б) при $m = 8$.

Программа 30. Вычисление $\arctg x$ по цепной дроби с $m = 8$

P2 7 × x^2 1 5 ÷ 1 3 ПП F8 3
 6 × 1 1 ПП F8 2 5 × 9 ПП F8
 1 6 × 7 ПП F8 9 × 5 ПП F8 4
 × 3 ПП F8 1 + ↑ F2 XY ÷ C/Π БП
 Р0 + ↑ F2 x^2 XY ÷ B/O

После занесения программы 30 в УЗП значение аргумента вводят в регистр X. Результат высвечивается на индикаторе. Погрешность вычислений по этой программе возрастает при увеличении аргумента. Так, при $x = 1$ значение $\arctg 1 = 0,7853972$ вычисляется с погрешностью $2 \cdot 10^{-6}$ рад, тогда как при $x = 3$ погрешность возрастает до $0,114$ рад. Время счета около 16 с. При необходимости можно дополнить программу переводом результата в градусы, записав коэффициент $180/\pi$ в регистр 3, заменив оператор С/Π фрагментом $\uparrow F3 \times C/Π$ и оператором F—.

Если аналитическая зависимость элементов цепной дроби от x и k достаточно проста, то можно составить программу с предварительным вычислением промежуточных результатов. Например, вычисления $\arctg x$ по цепной дроби (52б) можно реализовать программой 31, определяющей текущие значения $\beta_k = k^2x^2$ и $\alpha_k = 2k + 1$ с последующими вычислениями по формуле (116 б).

Программа 31. Вычисление $\operatorname{arctg} x$ по цепной дроби с заданным m

$$\begin{array}{ccccccccc} P2 & 0 & P4 & F3 & 2 & \times & 1 & - & \uparrow & F4 & + & P4 \\ F3 & 1 & - & P3 & x \neq 0 & F, & \uparrow & F2 & \times & x^2 & \uparrow & F4 \\ \div & P4 & \text{БП} & 0 & F2 & \uparrow & F4 & \div & C/P & \text{БП} & P0 \end{array}$$

Перед пуском программы 31 заданное число звеньев m заносят в регистр 3, а значение x — в индикаторный регистр. Результат высвечивается на индикаторе. Время вычислений по этой программе примерно равно $4m$. При $m = 2; 6; 12; 16$ программа 31 вычисляет $\operatorname{arctg} 2 = 0,8571428; 1,101231; 1,107130; 1,107148$ соответственно за время $t = 8; 23; 44; 59$ с.

Вычисление k -х подходящих дробей в соответствии с формулой (38) реализуется программой 32 ($\beta_k = P2, \alpha_k = P3, P_{k-1} = P4, P_{k-2} = P5, Q_{k-1} = P6, Q_{k-2} = P7$). Для вычисления очередного значения k -й подходящей дроби нажимается клавиша С/П (при первом пуске клавиши В/О и С/П).

Программа 32. Вычисление k -х подходящих дробей

$$\begin{array}{cccccccccc} F2 & \uparrow & F5 & \times & \rightarrow & F7 & \times & \rightarrow & F3 & \uparrow & F4 & P5 \\ \times & \rightarrow & F6 & P7 & \times & \uparrow & \leftarrow & \leftarrow & + & P6 & \rightarrow & \uparrow \\ \leftarrow & \leftarrow & + & P4 & \uparrow & F6 & \div & C/P & \text{БП} & P0 \end{array}$$

Так как значение цепной дроби лежит между двумя очередными значениями подходящих дробей, то для достижения требуемой точности вычисления по программе 32 следует повторять до совпадения требуемого числа значащих цифр в двух очередных результатах.

Дополнив программу 32 начальной частью, вычисляющей значения β_k и α_k по заданной функциональной зависимости, можно обеспечить вычисление функциональной цепной дроби с требуемой точностью. Примером может служить программа 33 для вычисления функции $\operatorname{arctg} x$.

Программа 33. Вычисление $\operatorname{arctg} x$ по подходящим дробям

$$\begin{array}{cccccccccc} P2 & P4 & 0 & P5 & 1 & P3 & P6 & P7 & F3 & 2 & + & P3 \\ 1 & - & 2 & \div & \uparrow & F2 & \times & x^2 & \uparrow & F5 & \times & \rightarrow \\ F7 & \times & \rightarrow & F3 & \uparrow & F4 & P5 & \times & \rightarrow & F6 & P7 & \times \\ \uparrow & \leftarrow & \leftarrow & + & P6 & \rightarrow & \uparrow & \leftarrow & \leftarrow & + & P4 & \uparrow \\ F6 & \div & \uparrow & F8 & XY & P8 & - & x = 0 & PXY & F8 & C/P \end{array}$$

Значение аргумента заносят в регистр X перед пуском программы клавишами В/О и С/П. Результат выводится на индикатор с точностью до семи значащих цифр. Затраты времени возрастают при увеличении аргумента и составляют 11 с при $\operatorname{arctg} 0,01 = 0,009999666$; 101 с при $\operatorname{arctg} 1 = 0,7853981$ и 331 с при $\operatorname{arctg} 5 = 1,373400$.

6. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Программы численного интегрирования (используемые самостоятельно или в качестве подпрограмм или фрагментов сложных программ для встроенного в ЭВМ или стандартного математического обеспечения) обычно составляют на основе формулы Симпсона (24а), которую можно записать

$$I = \int_a^b f(x)dx \approx \frac{h}{3} [f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \dots + 4f(b-h) + f(b)],$$

где $h = (b - a)/n$ — шаг интегрирования; n — четное число отрезков разбиения интервала интегрирования.

Эту формулу целесообразно использовать в качестве основной и для программирования тех микрокалькуляторов, во входных языках которых отсутствуют операторы численного интегрирования. Численное интегрирование по формуле Симпсона реализуется программой 34, перед пуском которой в регистры памяти вводят числа $a = P2$, $b = P3$, $n = P4$.

Программа 34. Численное интегрирование по формуле Симпсона

ПП	FC _x	P8	F3	↑	F2	P3	—	↑	F4	÷	P2
ПП	FC _x	1	ПП	P5	4	ПП	P5	2	БП	2	F8
3	÷	↑	F2	×	C/П	×	↑	F8	+	P8	F4
1	—	P4	$x \neq 0$	F÷	F3	↑	F2	+	P3	...	B/O

Заключительная часть программы 34, начиная с шага с адресом 75, предназначена для организации вычислений подынтегрального выражения по значению текущей переменной x , записываемой в регистр 3. Результат вычислений накапливается в регистре 8 и выводится на индикатор после выполнения программы. Например, для вычисления интеграла

$$I = \int_1^5 \frac{x^3}{x^4 + c} dx \quad (141)$$

по формуле Симпсона следует дополнить программу 34 арифметическим выражением, вычисляющим подынтегральное выражение. При $c = P5$ получим в этом случае программу 35, которая при $c = 16$ ($I = 0,9074539$) и $n = 4$ вычисляет $I = 0,9172362$ за 25 с, при $n = 8$ вычисляет $I = 0,9075659$ за 45 с и при $n = 64$ вычисляет $I = 0,907454$ за 300 с.

Программа 35. Вычисление интеграла (141)

ПП	FC _x	P8	F3	↑	F2	P3	—	↑	F4	÷	P2
ПП	FC _x	1	ПП	P5	4	ПП	P5	2	БП	2	F8
3	÷	↑	F2	×	C/П	×	↑	F8	+	P8	F4
1	—	P4	$x \neq 0$	F÷	F3	↑	F2	+	P3	F3	↑
x^2	×	P6	×	↑	F5	+	↑	F6	XY	÷	B/O

При численном интегрировании дробных функций необходимо учитывать, что обращение знаменателя в нуль на интервале интегрирования может вызвать переполнение (даже если все подынтегральное выражение остается ограниченным) в том случае, когда значение переменной интегрирования, обращающее знаменатель в нуль, совпадает с одним из значений $a + kh$ на краях отрезков разбиения, для которых вычисляется подынтегральная функция. Избежать переполнения можно следующими способами:

1. Если корни знаменателя лежат внутри интервала интегрирования, а подынтегральная функция ограничена на этом интервале, то следует выбирать число n разбиения интервала так, чтобы ни при каком $0 \leq k \leq n$ значение переменной интегрирования не совпадало с этими корнями.

2. Если подынтегральная функция на интервале интегрирования обращается в бесконечность, но есть основания считать интеграл сходящимся, то его следует преобразовать (например, интегрированием по частям) так, чтобы подынтегральное выражение на всем интервале интегрирования оставалось ограниченным.

3. Если у ограниченной на интервале интегрирования дробной функции знаменатель обращается в нуль на границах интервала, то проще всего вычислить значение интеграла в этих точках и записать его в программу, исключив автоматическое вычисление подынтегрального выражения в этих точках.

В качестве примера рассмотрим вычисление интегрального синуса

$$Si(x) = \int_0^x \frac{\sin t}{t} dt.$$

Так как нижний предел равен нулю, то программу 34 можно сократить на два шага, исключив вычисление разности пределов интегрирования. С учетом этого и особой точки при $t = 0$ составим программу 36 ($x = P3$, $n = P4$).

Программа 36. Вычисление интегрального синуса

1	P8	F3	\uparrow	F4	\div	P2	ПП	ВП	1	ПП	F4
4	ПП	F4	2	БП	FXУ	F8	3	\div	\uparrow	F2	\times
C/П	\times	\uparrow	F8	$+$	P8	F4	1	$-$	P4	$x \neq 0$	P3
F3	\uparrow	F2	$-$	P3	F3	\uparrow	sin	XУ	\div	B/O	

Контрольные результаты вычислений по этой программе, выводимые на индикатор: $Si(2) = 1,605497$ при $n = 4$ и времени счета 20 с; $Si(2) = 1,605418$ при $n = 8$ и времени счета 45 с, $Si(2) = 1,605413$ при $n = 16$ и времени счета 85 с и $Si(20) = 1,548241$ при $n = 100$ и времени счета 540 с.

Если длина незаполненной части программы 34 недостаточна для вычисления подынтегрального выражения, то можно предварительно вычислить шаг $h = (b - a)/n$ и занести его в один из регистров

памяти, что позволит сократить на семь шагов основную часть программы, увеличив соответственно с 13 до 20 число незанятых шагов. Можно также умножить сумму на $h/3$ после выполнения программы, что освободит еще пять шагов. В подобной упрощенной программе 37 ($a = P2$, $b = P3$, $n = P4$, $h = P5$) результат накапливается в регистре 8.

Программа 37. Суммирование по формуле Симпсона

ПП	5	P8	F2	P3	ПП	5	1	ПП	\times	4	ПП
\times	2	БП	PXY	C/P	\times	\uparrow	F8	$+$	P8	F4	1
—	P4	$x \neq 0$	F \times	F3	\uparrow	F5	$+$	P3	B/O

После окончания программы 37 для вычисления интеграла содержимое регистра 8 необходимо умножить на $h = P5$ и разделить на 3.

Программу интегрирования можно значительно сократить, используя формулу трапеций (23б) или прямоугольников, но результаты интегрирования в этих случаях будут отличаться большими погрешностями.

Полная погрешность интегрирования состоит не только из методической погрешности (погрешности ограничения), уменьшающейся при уменьшении шага h , но и погрешности округления (операционной погрешности), увеличивающейся при увеличении числа разбиений, т. е. при уменьшении h . Поэтому максимально достижимая точность результата соответствует оптимальному значению числа n или h , которое можно оценить согласно методике, рассмотренной в гл. 1.

При преобразовании подынтегральных дробных функций, обращающихся в бесконечность в некоторых точках интервала интегрирования, целесообразно одновременно попытаться представить несобственные интегралы обыкновенными.

Пример 51. Составить программу вычисления интеграла

$$p(x) = \frac{1}{\sqrt{2\pi^3}\sigma} \int_{-A}^A \frac{e^{-\frac{(x-y)^2}{2\sigma^2}}}{\sqrt{A^2 - y^2}} dy, \quad (142)$$

обращающегося в бесконечность на границах интервала интегрирования.

Заменой $y = A \cos z$ и нормированием переменных $t = x/\sigma$; $B = A/\sigma$ приведем заданный интеграл к виду

$$p(t) = \frac{1}{\sqrt{2\pi^3}} \int_0^\pi e^{-\frac{(t-B \cos z)^2}{2}} dz.$$

На основе программы 34 составим для вычисления этого интеграла программу 38 ($n = P4$, $B = P5$, $t = P6$, $1/3 \sqrt{2\pi^3} = P7$).

Программа 38 Вычисление интеграла (142)

```

0 ПП FCx P8 π P3 ↑ F4 ÷ P2 ПП 7
1 ПП , 4 ПП , 2 БП F2 F8 ↑ F7
× ↑ F2 × C/Π × ↑ F8 + P8 F4 1
— P4 x ≠ 0 3 F3 ↑ F2 — P3 F3 cos ↑
F5 × ↑ F6 — x² 2 ÷ | - | e^x B/O

```

В процессе вычислений значение $h = \Delta t$ заносится программой 38 в регистр 2, текущее значение переменной интегрирования $\pi = k\Delta t$ — в регистре 3, а в регистре 8 накапливается результат суммирования. Значение интеграла высвечивается на индикаторе после выполнения программы. Правильность ввода программы можно проверить по следующим контрольным данным: для $B = 1$ программа вычисляет $p(1) = 0,235893$ при $n = 8$ примерно за 80 с и $p(1) = 0,2358914$ при $n = 16$ примерно за 160 с.

На практике часто приходится интегрировать функции, заданные графическими или табличными моделями. Если отсчеты $y_i = y(x_i)$ функции заданы при постоянном шаге $h = \Delta x$ и их число четно, то для численного интегрирования можно использовать программу 39 ($h = P2$, $y_i = PX$).

Программа 39. Интегрирование табличных моделей по формуле парабол

```

↑ F8 x = 0 F4 XY P8 C/Π ↑ 4 × ↑ F8
+ P8 C/Π ↑ 2 × ↑ F8 + P8 C/Π БП
F1 + ↑ F2 × 3 ÷ P7 0 P8 F7. C/Π

```

При работе с программой 39 вводят первый отсчет y_0 и нажимают клавиши В/О и С/П, затем вводят y_i при $0 < i < n$ и нажимают только клавишу С/П (на индикаторе высвечиваются промежуточные результаты вычислений). После ввода последнего отсчета y_n и нажатия клавиш В/О и С/П выполняется программа и на индикаторе высвечивается значение интеграла.

Вычисления по формуле трапеций (236) реализованы в более короткой программе 40 ($h = P2$, $0 = P3$, $0 = PX$).

Программа 40. Интегрирование табличных моделей по формуле трапеций

```

P4 ↑ F3 + ↑ F2 × 2 ÷ P5 F4 P3
F5 ↑ F6 + P6 C/Π БП P0

```

После занесения программы 40 в УЗП в индикаторный регистр последовательно вводят отсчеты y_1, \dots, y_k, \dots с нажатием после каждого ввода клавиши С/П (после первого ввода y_0 нажимают клавиши В/О и С/П). Число узлов может быть любым, а после каждого выполнения программы высвечивается текущее значение интеграла. Время обработки одного отсчета около 4 с. При использовании занесенной в УЗП программы 40 для другой функции следует предварительно очистить регистры 6 и 3.

Контрольные результаты интегрирования табличной модели $y(x_i)$ с шагом $h = 0,2$ приведены в табл. 46.

Исходные данные	<i>i</i>	0	1	2	3	4	5	6	7	8
	<i>x_i</i>	0	0,2	0,4	0,6	0,8	1,0	1,2	1,4	1,6
	<i>y_i</i>	0,1	0,3	0,7	1,5	1,8	2,0	2,1	2,15	2,0
Программа 39	(0,1)	(1,3)	(2,7)	(8,7)	(12,3)	(20,3)	(24,5)	(33,1)	2,34	
Программа 40	0,01	0,05	0,15	0,37	0,7	1,08	1,49	1,915	2,33	

Оценка точности интегрирования при табличном задании функции лишена смысла, если значения функции между узлами модели не известны. Однако в тех случаях, когда табличной моделью представлена известная функциональная зависимость, формула парабол дает меньшую погрешность по сравнению с формулой трапеций или прямоугольников.

Вычисление несобственных интегралов вида

$$I = \int_a^{\infty} f(x) dx$$

обеспечивается программой 41 ($h = P2$, $a = P3$, $PX = I$), в которой арифметическое выражение, вычисляющее подынтегральную функцию, записано в заключительной части, начиная с шага 55.

Программа 41. Вычисление несобственных интегралов

0	P8	ПП	F/-/	1	ПП	Р÷	4	ПП	Р÷	2	БП
↑	F8	3	÷	↑	F2	×	C/P	×	↑	F8	+
P8	÷	sin	$x \neq 0$	F2	F3	↑	F2	+	P3	...	B/O

Если при подстановке нижнего предела подынтегральное выражение несобственного интеграла обращается в нуль, то в программе 41 следует заменить операторы F/-/; 1; 4; 2, расположенные по адресам 04; 05; 12 и 15, соответственно операторами; 4; 2 и 4.

В случаях, когда подынтегральное выражение сходится медленно, затраты времени на выполнение программы 41 можно уменьшить, увеличив шаг интегрирования, но погрешности результата при этом возрастут. Часто время интегрирования удается сократить, предварительно преобразовав подынтегральное выражение (например, интегрированием по частям) так, чтобы ускорить сходимость вычислительного процесса.

При интегрировании знакопеременных функций по программе 41 погрешность результата может резко возрасти, если текущее значение переменной интегрирования случайно совпадает с одним из корней подынтегрального выражения или достаточно близко к этому корню.

От указанных недостатков свободна программа 42, в которой выполняется последовательное интегрирование на интервалах $[x_k; x_k + 2h]$ с увеличением шага интегрирования по арифметической прогрессии на величину Δh . Критерием прекращения вычислений в этой программе выбрано условие максимальной точности (140), записываемое в данном случае как

$$\int_a^{x_k} f(x) dx + \int_{x_k}^{x_k+2h} f(x) dx = \int_a^{x_k} f(x) dx,$$

что соответствует равенству интеграла на интервале $x_k + 2h$ машинному нулю.

Программа 42. Вычисление несобственных интегралов с переменным шагом

ПП	PC _x	P8	ПП	6	4	ПП	F/-/-	↑	F8	+	XY
P8	F3	×	↑	F7	+	P7	÷	sin	x = 0	4	F7
3	÷	C/P	F3	↑	F2	+	P3	БП	0	×	↑
F8	+	P8	F4	↑	F3	+	P4	B/O

Результаты интегрирования при выполнении программы 42 ($h_0 = P3$, $\Delta h = P2$, $a = P4$) накапливаются в регистре 3 и после деления на 3 высвечиваются на индикаторе. Регистр 8 использован как вспомогательный для вычисления интеграла на интервале двойного текущего шага.

Для сравнения контрольные результаты вычисления интеграла

$$I = \int_0^{\infty} x e^{-x} dx$$

с точным значением $I = 1$ по программам 41 и 42, дополненным арифметическим выражением для вычисления подынтегральной функции, приведены в табл. 47.

По данным контрольного примера видно, что программа 42 имеет преимущества, однако она применима лишь при выполнении условия

$$\lim_{x \rightarrow \infty} |xf(x)| = 0,$$

Таблица 47

Программа	h	Δh	I	$t, \text{ с}$
41	1	—	0,9862110	110
	0,5	—	0,9990083	205
	0,2	—	0,9999737	445
42	1	1	0,9859554	75
	0,5	0,2	0,9984363	115
	0,2	0,2	0,9995294	150
	0,2	0,1	0,9998335	180
	0,05	0,05	0,9999719	240

которое не выполняется, например, для интегрального синуса.

7. ВЫЧИСЛЕНИЕ СПЕЦИАЛЬНЫХ ФУНКЦИЙ

Программируемый микрокалькулятор позволяет вычислять практически все специальные функции по заносимым в УЗП программам. Рассмотрим основные методы составления подобных программ на примере обратных тригонометрических функций, которые можно условно отнести к специальным в связи с отсутствием в используемом входном языке операторов вычисления этих функций.

Распространенный метод вычисления функций основан на их разложении в степенной ряд с последующим суммированием членов ряда до достижения требуемой точности. Подобный метод реализован программой 43 вычисления функции $\operatorname{arctg} x$ по ее разложению в ряд (50б). Условием окончания вычислений по этой программе выбран критерий максимальной точности (140). Значение аргумента x вводится в регистр X, где замещается результатом вычислений после выполнения программы.

Программа 43. Вычисление $\operatorname{arctg} x$ по разложению в степенной ряд

P2	x^2	/—/	P3	1	P4	P5	P8	F4	2	+	P4
F3	↑	F5	×	P5	↑	F4	÷	↑	F8	XY	+
P8	—	$x = 0$	PXY	F2	×	C/P	BП	P0			

Недостаток алгоритма, реализованного в программе 43, заключается в расходности ряда при $x > 1$, что приводит к резкому увеличению времени вычислений при приближении аргумента к единице. Например, если $\operatorname{arctg}(0,1) = 0,09966865$ программа 43 вычисляет примерно за 17 с, то для вычисления $\operatorname{arctg}(0,8)$ требуется более 100 с. Время счета по этой программе можно сократить, ограничивая число суммируемых членов ряда, но это значительно увеличивает погрешность. Так, если при $x = 1$ и числе $n = 29$ членов ряда погрешность результата составляет около 1° при времени счета около 92 с, то при $x = 1$ и $n = 287$ погрешность снижается до $0,1^\circ$, но время счета возрастает до 911 с.

Значительно лучшие результаты достигаются при разложении функции $\operatorname{arctg} x$ в цепную дробь. Программа 31 вычисляет $\operatorname{arctg} 1$ с погрешностью менее $1,8 \cdot 10^{-8}$ рад ($0,1^\circ$) при пяти звеньях дроби за 19 с, а для получения результата с предельной точностью до семи верных знаков требуется 11 звеньев (время счета около 42 с). Ограничение аргумента интервалом $-1 < x < 1$ не препятствует определению рассматриваемой функции при $x = 1$, так как можно воспользоваться соотношением (51) и перейти к аргументу $1/x$. Однако в этом случае длина программы увеличивается вследствие дополнения ее арифметическими выражениями, изменяющими ход вычислений в зависимости от значения аргумента.

Один из возможных способов вычисления специальных функций основан на численном интегрировании порождающих их дифферен-

циальных уравнений, часто описываемых достаточно простыми формулами. Например, вычисление функции

$$\operatorname{arctg} x = \int_0^x \frac{dt}{1+t^2}$$

можно организовать в виде программы 44, учитывающей равенство единице подынтегрального выражения при $t = 0$. Четное число n отрезков разбиения интервала интегрирования перед пуском программы (клавишами В/О и С/П) заносится в регистр 4, а значение x — в регистр X.

Программа 44. Вычисление $\operatorname{arctg} x$ по формуле парабол

P3	↑	F4	÷	P2	1	P8	F3	ПП	P7	1	ПП
P,	4	ПП	P,	2	БП	XY	F8	3	÷	↑	F2
×	C/П	×	↑	F8	+	P8	F4	1	—	P4	$x \neq 0$
F3	F3	↑	F2	—	P3	x^2	1	+	1/x	B/O	

Для достижения высокой точности при увеличении x необходимо увеличивать число n отрезков интегрирования. Например, при $x = 2$ и $n = 8$ результат $\operatorname{arctg} 2 = 1,107140$ (точное значение 1,107149) программа 44 вычисляет за 32 с, а $\operatorname{arctg} 10 = 1,469182$ (точное значение 1,471128) при $n = 20$ за 80 с.

Обратные тригонометрические функции можно вычислять также при помощи решения трансцендентного уравнения вида $f(y) = x$. Например, решение уравнения $\sin y = x$ относительно $y = \arcsin x$ реализовано в программе 45 ($x = RX$) по второй из формул (56), которая в данном случае при $a = x$ принимает вид

$$y_{i+1} = \frac{xy_i}{\sin y_i}; \quad i = 0, 1, \dots$$

С учетом характера изменения функции в интервале $0 < x < \pi/2$ в программе 45 в качестве начального приближения выбрано $y_0 = x$. Заданное значение аргумента вводится в индикаторный регистр, где замещается результатом после выполнения программы.

Программа 45. Вычисление $\arcsin x$ методом хорд

P2	P3	F3	↑	sin	÷	↑	F2	×	↑	F3	XY
P3	÷	1	—	$x = 0$	P↑	F3	C/П	БП	P0		

Время счета по программе 45 увеличивается по мере приближения аргумента к единице — если вычисление $\arcsin 0,1 = 0,1001676$ длится 18 с, то для $\arcsin 0,9 = 1,119769$ требуется 86 с.

Значительно лучшие результаты получаются при решении уравнения $\sin y = x$ методом касательных по формуле (57), принимающей вид

$$y_{i+1} = y_i + \frac{x - \sin y_i}{\cos y_i}; \quad i = 0, 1, \dots$$

Вычисления по этой формуле при $y_0 = x$ реализуются программой 46.

Программа 46. Вычисление $\arcsin x$ методом касательных

P2 P3 F3 e^{ix} P4 F2 — ↑ F4 ÷ ↑ F3
 X-Y — P3 — $x = 0$ P↑ F3 C/P БП Р0

Программа 46 ($x = PX$) вычисляет $\arcsin 0,5 = 0,5235987$ за 13 с, а $\arcsin 1 = 1,570466$ за 60 с. Таким образом, программа 46 при одинаковой длине значительно превосходит программу 45 в отношении затрат времени на вычисления. При $x \leq 0,99$ программа 46 вычисляет результат с семью верными знаками, при большем аргументе погрешность увеличивается, но в любом случае результат имеет не менее пяти верных знаков.

Возможность сокращения затрат времени или длины программы часто можно обнаружить, изучив свойства вычисляемой функции, например, используя формулы

$$\sin x/2 = \sqrt{(1 - \cos x)/2}; \quad \cos x/2 = \sqrt{(1 + \cos x)/2}; \\ \cos x = \pm \sqrt{1 - \sin^2 x},$$

можно заменить исходное выражение $y = \arcsin x$ соотношением

$$y = 8 \arcsin \sqrt{(1 - \sqrt{(1 + \sqrt{1 - x^2})/2})/2}/2. \quad (143)$$

В этом случае диапазон значений аргумента арксинуса ограничивается интервалом $[0; 0,1950902]$, в пределах которого все рассмотренные программы обеспечивают достаточно малое время счета. Однако при практическом использовании подобного алгоритма оказывается, что для малых значений аргумента погрешность результата возрастает в связи с погрешностями округлений при многократном извлечении корней. Можно избежать этой потери точности, построив разветвленную программу, которая вычисляет результат при $x < 0,2$ любым из рассмотренных ранее способов, а при $x \geq 0,2$ — согласно формуле приведения (143). Подобный подход реализован в программе 47, вычисляющей $\arcsin x$ методом хорд. Значение аргумента вводится в регистр X и после пуска программы клавиши B/O и C/P на индикаторе высвечивается результат в радианах. При дополнительном нажатии клавиши C/P высвечивается результат в градусах.

Для интервала $x < 0,2$ результат вычислений по программе 47 имеет шесть верных знаков уже после двух итераций. Программа вычисляет $\arcsin 0,1 = 0,1001674$ за 6 с, а $\arcsin 1 = 1,570793$ за 11 с. Таким образом, время вычислений по этой программе достаточно мало, а точность результата достаточна для любых практических приложений.

Программа 47. Вычисление $\arcsin x$ по формуле приведения

P2 1 P3 F2 5 1/x — $x \geq 0$ F/-/ 8 P3 F2
 x^2 1 XY — $\sqrt{1 - x^2}$ 1 + 2 ÷ $\sqrt{1 - x^2}$ 1 +
 2 ÷ $\sqrt{1 - x^2}$ 1 XY — 2 ÷ $\sqrt{1 - x^2}$ P2 F2 ↑
 \sin ÷ \times ↑ sin ÷ ↑ F2 \times ↑ F3 \times
 C/P ↑ π ÷ 1 8 0 × C/P

Используя формулу приведения, подобную выражению (143), можно составить аналогичную разветвленную программу и для вычисления функции $\operatorname{arctg} x$, но существенная нелинейность последней приводит к большим затратам времени на ее вычисление. Например, применяя метод хорд можно вычислить $\operatorname{arctg} 1$ с шестью верными знаками только после 15 итераций, а при использовании метода касательных для этого требуется не менее четырех итераций.

Эффективный способ составления быстродействующих программ вычисления функций заключается в использовании приближенных выражений, аппроксимирующих функцию с заданной точностью. В отличие от рассмотренных ранее программ (где погрешность вычислений определялась в основном конечной разрядностью операндов, а не выбором алгоритма), в программах, реализующих приближенные выражения, точность результата в основном зависит от удачного выбора аппроксимирующей формулы. Усложнение аппроксимирующих выражений удлиняет программы вычисления, что затрудняет их использование в качестве фрагментов более общих программ.

Программы 48—51 вычисляют функцию $\operatorname{arctg} x$ в интервале $\pm 90^\circ$ по различным приближенным выражениям, указанным в табл. 48, и отличаются числом операторов и точностью результата.

Таблица 48

Программа	Алгоритм вычислений	Число операторов в программе	Погрешность результата, град	Время вычислений, с	Интервал допустимых значений	Представление результата и его интервал
30	По цепной дроби с $m = 8$	56	<0,00012	16	± 1	$\pm \pi/4$
31	По цепной дроби с заданным m	35	Зависит от m	Зависит от m	Зависит от m	Радианы
33	По подходящим дробям	59	<0,00009°	Зависит от x	$\pm \infty$	$\pm \pi$
48	$57(1 - 0,3x^2 \times (1 - 0,3x^2))x$	40	$\leq 0,14^\circ$	6	$\pm \infty$	$\pm 90^\circ$
49	$\frac{57x}{\sqrt{1 + 0,607x^2}}$	41	$\leq 0,049^\circ$	6	$\pm \infty$	$\pm 90^\circ$
50	$\frac{81x}{\sqrt{1 + \sqrt{1 + x^2} + 0,81x^2}}$	26	$< 0,25^\circ$	5	$\pm \infty$	$\pm 90^\circ$
51	$\frac{80,67x}{\sqrt{1 + \sqrt{1 + x^2} + 0,805x^2}}$	31	$\leq 0,09^\circ$	5	$\pm \infty$	$\pm 90^\circ$

Программа 48. Вычисление $\operatorname{arctg} x$ с погрешностью менее $0,14^\circ$

$$\begin{array}{l} \text{P2 } x^2 \quad 1 \quad - \quad x \geqslant 0 \quad \text{XY} \quad \text{F2 } 1/x \quad /-/- \quad \text{P2 } 9 \quad 0 \\ \text{P3 } F2 \quad x \geqslant 0 \quad P \div \quad F3 \quad /-/- \quad P3 \quad F2 \quad x^2 \quad 0 \quad , \quad 3 \\ \times \quad 1 \quad - \quad \times \quad 1 \quad + \quad 5 \quad 7 \quad \times \quad \uparrow \quad \text{F2 } \times \\ \uparrow \quad F3 \quad + \quad C/\Pi \end{array}$$

Программа 49. Вычисление $\operatorname{arctg} x$ с погрешностью менее $0,049^\circ$

$$\begin{array}{l} \text{P2 } x^2 \quad 2 \quad - \quad x \geqslant 0 \quad \text{XY} \quad \text{F2 } 1/x \quad /-/- \quad \text{P2 } 9 \quad 0 \\ \text{P3 } F2 \quad x \geqslant 0 \quad P \div \quad F3 \quad /-/- \quad P3 \quad F2 \quad x^2 \quad 0 \quad , \quad 6 \\ 0 \quad 7 \quad \times \quad 1 \quad + \quad V^- \quad \uparrow \quad \text{F2 } XY \quad \div \quad 5 \quad 7 \\ \times \quad \uparrow \quad F3 \quad + \quad C/\Pi \end{array}$$

Программа 50. Вычисление $\operatorname{arctg} x$ с погрешностью менее $0,25^\circ$

$$\begin{array}{l} \text{P2 } 0 \quad , \quad 9 \quad \times \quad x^2 \quad 1 \quad + \quad \text{P3 } F2 \quad x^2 \quad 1 \\ + \quad V^- \quad \uparrow \quad F3 \quad + \quad V^- \quad \uparrow \quad \text{F2 } XY \quad \div \quad 8 \quad 1 \\ \times \quad C/\Pi \end{array}$$

Программа 51. Вычисление $\operatorname{arctg} x$ с погрешностью менее $0,09^\circ$

$$\begin{array}{l} \text{P2 } x^2 \quad 0 \quad , \quad 8 \quad 0 \quad 5 \quad \times \quad 1 \quad + \quad \text{P3 } F2 \\ x^2 \quad 1 \quad + \quad V^- \quad \uparrow \quad F3 \quad + \quad V^- \quad \uparrow \quad \text{F2 } XY \quad \div \\ 8 \quad 0 \quad , \quad 6 \quad 7 \quad \times \quad C/\Pi \end{array}$$

Во всех этих программах значение аргумента заносится в регистр X, а результат вычислений выводится на индикатор в градусах. Дополняя программы фрагментом $\uparrow \pi \times 18 \quad 0 \div$, можно дополнительно перевести результат в радианы. Для сравнения в табл. 48 приведены также основные характеристики ранее составленных программ вычисления $\operatorname{arctg} x$ с относительно малым временем счета. Для большинства практических задач наиболее удобна программа 51, которая при среднем числе операторов обеспечивает приемлемую точность результатов.

Вычисление функции $\operatorname{arcsin} x$ с погрешностью менее $0,09^\circ$ обеспечивается программой 52 за время около 6с.

Программа 52. Вычисление $\operatorname{arcsin} x$ с погрешностью менее $0,09^\circ$

$$\begin{array}{l} \text{P2 } x^2 \quad 0 \quad , \quad 1 \quad 9 \quad 5 \quad \times \quad \text{P3 } F2 \quad x^2 \quad 1 \\ XY \quad - \quad V^- \quad 1 \quad + \quad \uparrow \quad F3 \quad - \quad V^- \quad \uparrow \quad \text{F2 } XY \\ \div \quad 8 \quad 0 \quad , \quad 6 \quad 7 \quad \times \quad C/\Pi \quad \text{БП} \quad \text{Р0} \end{array}$$

В универсальной программе 53 вычисления всех обратных тригонометрических функций в качестве фрагмента использована программа 51 для вычисления $\operatorname{arctg} x$. Остальные функции вычисляются в соответствии с формулами (51).

Программа 53. Вычисление обратных тригонометрических функций

P2	x^2	1	—	$x < 0$	\times	/—/	V^-	\uparrow	F2	XY	÷
ПП	F4	P4	—	P5	F2	ПП	F4	P6	—	P7	F6
C/П	P8	x^2	0	,	8	0	5	\times	1	+	P3
F8	x^2	1	+	V^-	\uparrow	F3	\uparrow	V^-	\uparrow	F8	XY
÷	8	0	,	6	7	\times	9	0	XY	B/O	

Перед пуском программы 53 (нажатием клавиши В/О и С/П) значение аргумента заносится в регистр X. После выполнения программы результаты оказываются занесенными в регистры P4 = $\arcsin x$, P5 = $\arccos x$, P6 = PX = $\arctg x$, P7 = $\operatorname{arcctg} x$. Результат выводится для $\arcsin x$ и $\arctg x$ в интервале $\pm 90^\circ$, а для $\arccos x$ и $\operatorname{arcctg} x$ — в интервале $0; 180^\circ$. При $x \geq 1$ функции arssin x и arccos x не вычисляются, и содержание регистров 4 и 5 не изменяется. Контрольные результаты для вычислений по программе 53 приведены в табл. 49.

Таблица 49

x	0,1	0,5	-0,5	0,999999	1	10
$\arcsin x$	5,714185	29,92069	-29,92069	89,83228	—	—
$\arccos x$	84,28581	60,07930	119,9206	0,16772	—	—
$\arctg x$	5,685728	26,48531	-26,48531	44,96109	44,96112	84,31079
$\operatorname{arcctg} x$	84,31427	63,51468	116,4353	45,03890	45,03887	5,68921
t, с	16	14	15	0	9	9

Рассмотренные методы составления программ пригодны для программирования вычислений и других функций, но при вычислении гиперболических функций целесообразно использовать оператор e^x , имеющийся во входных языках практически всех программируемых микрокалькуляторов.

Гиперболические функции вещественного аргумента x вычисляются при помощи следующих простых арифметических выражений: sh x по программе

$$e^x \uparrow 1/x - 2 \div \text{С/П БП Р0};$$

ch x — по программе

$$e^x \uparrow 1/x + 2 \div \text{С/П БП Р0};$$

th x — по программе

$$\uparrow 2 \times e^x 1 - 2 + \div \text{С/П БП Р0}.$$

При использовании этих программ значение аргумента вводится в индикаторный регистр, где замещается результатом выполнения программы.

Гиперболический синус комплексного аргумента

$$\operatorname{sh}(x + jy) = \operatorname{sh} x \cos y + j \operatorname{ch} x \sin y$$

можно вычислить по программе 54.

Программа 54. Вычисление $\operatorname{sh}(x + jy)$

P2	XY	e^x	\uparrow	$1/x$	P3	—	2	\div	P4	\uparrow	F3
+	\uparrow	F2	\sin	\times	P3	F4	\uparrow	F2	\cos	\times	\uparrow
F3	C/P	BП	P0								

Перед пуском программы 54 вещественную часть аргумента заносят оператором \uparrow в регистр Y, а мнимую вводят в регистр X. Аналогично размещены вещественная и мнимая части результата после выполнения программы и для вызова вещественной части достаточно нажать клавишу $X \Leftrightarrow Y$.

Для вычисления гиперболического косинуса $\operatorname{ch}(x + jy) = \operatorname{ch} x \cos y + j \operatorname{sh} x \sin y$ в программе 54 достаточно поменять местами операторы \sin и \cos по адресам 24 и 34 и дополнить программу оператором $X \Leftrightarrow Y$ по адресу 42 перед оператором С/П.

Значение гиперболического тангенса комплексного аргумента

$$\operatorname{th}(x + jy) = \frac{e^{2x} - e^{-2x} + 2j \sin 2y}{e^{2x} + e^{-2x} + 2 \cos 2y}$$

можно вычислить с помощью программы 55, порядок работы с которой аналогичен порядку работы с программой 54.

Программа 55. Вычисление $\operatorname{th}(x + jy)$

P2	XY	2	\times	e^x	\uparrow	$1/x$	P3	—	2	\div	P4
\uparrow	F3	$+$	P3	F2	\times	P2	\cos	\uparrow	F3	$+$	
P3	\uparrow	F2	\sin	XY	\div	P2	F4	\uparrow	F3	\div	\uparrow
F2	C/P	BП	P0								

Для проверки правильности занесения в УЗП программ 54 и 55 (выполняемых менее чем за 12 с) можно воспользоваться следующими контрольными результатами: $\operatorname{sh}(1 + j) = 0,6349638 + j1,298457$; $\operatorname{ch}(1 + j) = 0,8337300 + j0,9888974$; $\operatorname{th}(1 + j) = 1,083923 + j0,2717526$.

Интегральный синус целесообразно вычислять по разложению в степенной ряд (54), реализуемому программой 56.

Программа 56. Вычисление интегрального синуса

P2	P7	P8	x^2	2	\div	/—/	P3	0	P4	F4	1
+	P4	2	\times	1	$+$	x^2	P5	F4	2	\times	1
—	\uparrow	F3	\times	\uparrow	F7	\times	\uparrow	F4	\div	\uparrow	F5
\div	P7	\uparrow	XY	$+$	P8	$-$	$x=0$	FXY	F8	C/P	

При использовании этой программы значение аргумента заносится в регистр X и вычисления продолжаются до выполнения условия (140). Результат записывается в регистры 8 и X, а в регистр 4

заносится число $k + 1$, на единицу превышающее число членов ряда, суммируемых для получения результата.

Для $x > 5$ точность вычислений по программе 56 вследствие погрешностей округления резко уменьшается. В этом случае можно воспользоваться программой 36 для численного интегрирования, позволяющей получить результат с семью верными знаками при достаточно большом числе разбиений, но время счета чрезмерно возрастает. Поэтому при $x > 8$ целесообразно использовать асимптотический ряд

$$Si(x) \approx \frac{\pi}{2} - \frac{\cos x}{x} \sum_{k=0}^{\infty} (-1)^k \frac{(2k)!}{x^{2k}} - \frac{\sin x}{x} \sum_{k=0}^{\infty} (-1)^k \frac{(2k+1)!}{x^{2k+1}}.$$

По программе 57 вычисляются четыре члена такого ряда, причем для компенсации остаточной погрешности коэффициенты последних членов такого усеченного ряда уменьшены

$$Si(x) \approx \frac{\pi}{2} - \frac{\cos x}{x} \left(1 - \frac{2}{x^2} \left(1 - \frac{12}{x^2} \left(1 - \frac{12}{x^2} \right) \right) \right) - \\ - \frac{\sin x}{x^2} \left(1 - \frac{6}{x^2} \left(1 - \frac{20}{x^2} \left(1 - \frac{20}{x^2} \right) \right) \right).$$

Программа 57. Вычисление интегрального синуса при $x \rightarrow \infty$

P2	x^2	1/x	P3	1	2	\times	1	-	\times	1	+
4	\times	\uparrow	F3	\times	2	-	\uparrow	F2	\cos	\times	\uparrow
F2	\div	P4	F3	2	0	\times	1	-	\times	1	+
1	2	\times	\uparrow	F3	\times	2	-	\uparrow	F2	\sin	\times
\uparrow	F3	\times	\uparrow	F4	$+$	\uparrow	π	$+$	2	\div	C/P

Для сравнения укажем, что на вычисление $Si(20) = 1,5482$ по программе 56 затрачивается 175 с (с результатом 1,5966622), по программе 36 затрачивается 540 с при $n = 100$ (с результатом 1,548241), а по программе 57 результат $Si(20) = 1,548241$ вычисляется за 15 с. При вычислении $Si(0,2)$, когда программа 57 неприменима, время счета $Si(0,2) = 0,1995560$ по программе 56 составляет 18 с, а по программе 36 при $n = 4$ результат $Si(0,2) = 0,1995561$ вычисляется за 19 с.

Интеграл вероятности $\Phi(x)$ вычисляется по программе 58 разложением в степенной ряд (55а). Перед пуском этой программы значение аргумента вводят в регистр X, результат вычислений после выполнения программы заносится в регистры 8 и X, а в регистр 4 записывается число суммированных членов ряда.

Программа 58. Вычисление интеграла вероятности по разложению в ряд

P2	x^2	2	\div	/-	P3	2	\uparrow	π	\div	V	\uparrow
F2	\times	P7	P8	0	P4	F4	1	$+$	P4	2	\times
1	$+$	P5	F4	2	\times	1	-	\uparrow	F3	\times	\uparrow
F4	\div	\uparrow	F5	\div	\uparrow	F7	\times	P7	\uparrow	F8	XY
$+$	P8	$-$	$x=0$	P3	F8	C/P	BП	P0			

При $x = 4$ программа 58 обеспечивает результат с пятью верными цифрами после запятой, при меньших значениях аргумента точность повышается. По этой программе $\Phi(0,1) = 0,07965567$ и $\Phi(1) = 0,6826894$ вычисляются со всеми верными знаками за 20 и 45 с соответственно.

При больших значениях аргумента интеграл вероятности можно вычислять по программе 59 ($x = P2$, $n = P3$). Точность результата зависит от числа n разбиений интервала интегрирования.

Программа 59. Численное интегрирование интеграла вероятности

$$\begin{array}{ccccccccc} P2 & \uparrow & F3 & \div & P4 & F2 & \text{ПП} & F - 1 + & P8 & \text{ПП} \\ F7 & 4 & \times & \uparrow & F8 & + & P8 & \text{ПП} & F7 & 2 \times \uparrow \\ F8 & + & P8 & \text{БП} & XY & F8 & 3 & \div & \uparrow & F4 \times 2 \\ V - & \times & \uparrow & \pi & V - & \div & C/\Pi & F3 & 1 - & x \neq 0 \\ P3 & \uparrow & F4 & \times & x^2 & 2 & \div & / - / & e^x & B/O \end{array}$$

Для значений $x > 4,5$ целесообразно использовать программу 60 вычисления интеграла вероятности по трем членам асимптотического ряда (55б), дающими результат с семью верными знаками.

Программа 60. Вычисление $\Phi(x)$ по асимптотическому разложению

$$\begin{array}{ccccccccc} P2 & x^2 & P3 & 2 & \div & / - / & e^x & 2 & V \times \uparrow \pi \\ V - & \div & \uparrow & F2 & \div & P2 & 3 & \uparrow & F3 \div 1 XY \\ \uparrow & F3 & \div & 1 & - & \uparrow & F2 \times 1 & + & C/\Pi \\ \text{БП} & P0 & & & & & & & \end{array}$$

Значение $\Phi(4) = 0,9951122$ при $n = 4$ вычисляется по программе 59 за 18 с, $\Phi(4) = 0,9999364$ при $n = 20$ за 95 с, $\Phi(4) = 0,9999366$ при $n = 40$ за 190 с, а по программе 60 значение $\Phi(4) = 0,9999365$ вычисляется менее чем за 10 с.

Функции Бесселя можно вычислять по разложению в степенной ряд (53а) с помощью программы 61 ($x = P2$, $n = P3$), заносящей результат в регистры 8 и X. Эта программа обеспечивает высокую точность вычислений при $x \leq 10$, $n \geq x$.

Программа 61. Вычисление функций Бесселя по разложению в ряд

$$\begin{array}{ccccccccc} C_x & P4 & 1 & P6 & F3 & x \neq 0 & \times & P8 & \uparrow & F6 & \times & P6 \\ F8 & 1 & - & x = 0 & F1 & F2 & 2 & \div & \uparrow & x^2 & P2 & F3 \\ XY & x^y & \uparrow & F6 & \div & P7 & P8 & F4 & 1 & + & P4 & \uparrow \\ F3 & + & \times & \uparrow & F2 & \div & \uparrow & F7 & XY & \div & / - / & P7 \\ \uparrow & F8 & XY & + & P8 & - & x = 0 & F5 & F8 & C/\Pi & \text{БП} & P0 \end{array}$$

По программе 61 вычисляется $J_0(2) = 0,2238907$ при восьми членах ряда за 45 с, $J_4(2) = 0,03399571$ при шести членах ряда за 33 с и $J_{30}(20) = 0,000124014$ при 19 членах ряда за 125 с.

Для вычисления модифицированных функций Бесселя (53б) достаточно в программе 61 исключить оператор изменения знака с адресом 75 (в занесенной в УЗП программе его можно заменить оператором НОП).

Используя соотношения

$$J_n(x) = \begin{cases} \frac{2}{\pi} \int_0^{\pi/2} \cos(x \sin \varphi) \cos n \varphi d\varphi & \text{для четных } n \\ \frac{2}{\pi} \int_0^{\pi/2} \sin(x \sin \varphi) \sin n \varphi d\varphi & \text{для нечетных } n \end{cases}$$

нетрудно составить алгоритм вычисления функций Бесселя численным интегрированием. Перед пуском составленной по такому алгоритму программы 62 для функций Бесселя четного порядка число разбиений (четное) интервала интегрирования, порядок функции и значение аргумента вводят соответственно в регистры 4, 7 и 8.

Программа 62. Вычисление функций Бесселя интегрированием

1	P6	π	2	\div	P3	\uparrow	F4	P5	\div	P2	ПП
PC _x	1	ПП	,	4	ПП	,	2	БП	P \times	F6	3
\div	\uparrow	F5	\div	C/P	\times	\uparrow	F6	$+$	P6	F4	1
—	P4	$x \neq 0$	F \div	\uparrow	F2	\times	P3	F3	\uparrow	F7	\times
cos	F3	F8	XY	sin	\times	cos	\uparrow	F3	\times	B/O	

Результат вычислений по программе 62 выводится на индикатор. При нечетном порядке вычисляемой функции Бесселя операторы cos, записанные в программе по адресам 81 и 91, необходимо заменить операторами sin, а первый оператор 1 — оператором C_x. Точность результата до шести верных знаков включительно обеспечивается программой 62 при выборе числа разбиений интервала интегрирования не менее чем в 2 раза превосходящим меньшее из чисел n или x .

Например, при шести разбиениях результат вычислений по этой программе $J_4(2) = 0,03398833$, а при 8 и 10 — соответственно $J_4(2) = 0,03399575$ и $J_4(2) = 0,03399577$.

Функции Бесселя высших порядков можно вычислять по формуле (53в), реализуемой программой 63, перед пуском которой значения $J_0(x)$, $J_1(x)$ и x вводят соответственно в регистры 2, 3 и X. Вычисленный результат хранится в регистрах 3 и X.

Программа 63. Вычисление функций Бесселя по $J_0(x)$ и $J_1(x)$

P8	1	P4	2	\times	\uparrow	F8	\div	\uparrow	F3	\times	\uparrow
F2	—	\uparrow	F3	P2	XY	P3	C/P	F4	1	$+$	БП
P	\uparrow										

Время счета по программе 63 составляет несколько секунд для функции очередного порядка, но с его увеличением точность вычислений уменьшается. Например, при исходных значениях $J_0(7) = 0,3001$ и $J_1(7) = -0,004683$ программа 63 обеспечивает вычисление $J_2(7) = -0,301438 \dots J_{11}(7) = 0,008336732; J_{12}(7) = 0,002659603;$

$J_{13}(7) = 0,0007819017 \dots$ при значениях с четырьмя верными цифрами $J_2(7) = -0,3014; \dots J_{11}(7) = 0,008335; J_{12}(7) = 0,002656; J_{13}(7) = 0,0007702.$

При вычислениях гамма-функции аппроксимация степенным рядом приводит к чрезмерным затратам времени. Численное интегрирование для получения результата с точностью до шести — восьми верных цифр также требует десятков минут. Поэтому наиболее приемлемой для программируемого микрокалькулятора является формула Стирлинга

$$\Gamma(z) = \sqrt{\frac{2\pi}{z}} e^{-z} z^z H(z),$$

где $H(z) = 1 + \frac{1}{12z} + \frac{1}{288z^2} - \frac{139}{51840z^3} - \frac{571}{2488320z^4}.$

В реализующей эту формулу программе 64 ($z = PX$) использовано приближение

$$H(z) \approx 1 + \frac{1}{12z} + \frac{1}{288z^2} - \frac{0,7}{288z^3}.$$

Точность вычислений по этой программе около пяти верных цифр.

Программа 64. Вычисление гамма-функции

P2	1	+	\times	P3	F2	2	+	P4	1	e^x	\div
x^y	\uparrow	F3	\div	P3	π	2	\times	\uparrow	F4	\div	V
\uparrow	F3	\times	P3	0	,	7	\uparrow	F4	\div	1	XY
\uparrow	2	4	\div	\uparrow	F4	\div	1	\uparrow	1	2	\div
\uparrow	F4	\div	1	\uparrow	F3	\times	C/P				

Для повышения точности результата при малых значениях аргумента в программе 64 использовано соотношение $\Gamma(z) = \Gamma(z + 2)/z(z + 1)$. Учитывая, что $n! = \Gamma(n + 1)$, по программе 64 можно вычислять факториалы больших чисел. Однако она применима только при положительных значениях аргумента и при $z < 0$ следует использовать соотношение $\Gamma(-z) = -\pi/[z\Gamma(z) \sin z]$, определив по программе 64 значение $\Gamma(z)$ и вручную или с помощью дополнительной программы вычисляя требуемый результат согласно приведенному соотношению.

Правильность ввода программы 64 в УЗП можно проверить по контрольным значениям $\Gamma(1) = 1,000000; \Gamma(8) = 5039,945$.

Эллиптические (неполные) интегралы первого рода

$$F(k, \varphi) = \int_0^\varphi \frac{d\psi}{\sqrt{1 - k^2 \sin^2 \psi}}$$

и второго рода

$$E(k, \varphi) = \int_0^\varphi V \sqrt{1 - k^2 \sin^2 \psi} d\psi,$$

можно вычислить методами численного интегрирования. Этот способ использован в программе 65 для вычисления эллиптических интегралов первого рода (четное $n = P4$, $k = P7$, $\varphi = P8$). Результат выводится на индикатор.

Программа 65. Вычисление эллиптических интегралов

1	P6	F8	P3	↑	F4	÷	P2	ПП	ВП	1	ПП
P,	4	ПП	P,	2	БП	XY	F6	3	÷	↑	F2
×	C/П	×	↑	F6	+	P6	F4	1	—	P4	$x \neq 0$
F3	↑	F2	×	P3	F3	sin	↑	F7	×	x^2	1
XY	—	V	1/x	B/O							

Для вычисления эллиптических интегралов второго рода в программе 65 достаточно исключить оператор $1/x$ по адресу 84. Программа 65 вычисляет $F(0,984807; \pi/2) = 3,153361$ при $n = 40$ за 250 с, $F(0,766044; 0,8726646) = 0,9400749$ при $n = 10$ за 60 с. Вычисление интегралов Френеля

$$C(x) = \int_0^x \cos\left(\frac{\pi}{2} t^2\right) dt; \quad S(x) = \int_0^x \sin\left(\frac{\pi}{2} t^2\right) dt$$

по их разложениям в степенные ряды

$$C(x) = \sum_{n=0}^{\infty} (-1)^n \left(\frac{\pi}{2}\right)^{2n} \frac{x^{4n+1}}{(2n)!(4n+1)},$$

$$S(x) = \sum_{n=0}^{\infty} (-1)^n \left(\frac{\pi}{2}\right)^{2n+1} \frac{x^{4n+3}}{(2n+1)!(4n+3)}$$

выполняется по программам 66 и 67 соответственно ($x = PX$).

Программа 66. Вычисление $C(x)$ с помощью степенного ряда

P8	P7	x^2	x^2	↑	π	x^2	×	4	÷	P2	0
P3	F3	1	+	P3	2	×	1	—	×	→	F3
4	×	3	—	4	+	÷	↑	←	÷	↑	F2
×	↑	F7	×	/—/	P7	↑	F8	XY	+	P8	—
$x = 0$	F2	F8	C/П	БП	P0						

Программа 67. Вычисление $S(x)$ с помощью степенного ряда

P8	x^2	↑	π	×	2	÷	↑	x^2	P2	F8	×
3	÷	P7	P8	0	P3	F3	1	+	P3	2	×
1	+	×	→	F3	4	×	1	—	4	+	÷
↑	←	÷	↑	F2	×	↑	F7	×	/—/	P7	↑
F8	XY	+	P8	—	$x = 0$	P3	F8	C/П	БП	P0	

По программе $C(1) = 0,7798934$ вычисляется примерно за 40 с и $S(1) = 0,4382591$ за 50 с. Для оценки погрешности вычислений по методике, рассмотренной в гл. 2, определим номер $k \approx \pi x^2/4$ максимального по величине члена ряда $C(x)$

$$|u_{k\max}| \approx \frac{1}{\pi^2 x^2} e^{\pi x^2/2},$$

что позволяет оценить верхнюю границу применимости программ 66 и 67. Если значение аргумента повышает этот предел, то для вычисления точных значений $C(x)$ и $S(x)$ можно воспользоваться программой 68 ($n = P2$, $x = PX$), реализующей формулу (24а). При $x = 1$ и $n = 8$ значение $C(x) = 0,7799349$ вычисляется по этой программе за 50 с, а при $n = 32$ для вычисления $C(x) = 0,7798935$ требуется около 180 с. Для вычисления $S(x)$ в программе 68 достаточно оператор \cos по адресу 73 заменить оператором \sin .

Программа 68. Вычисление $C(x)$ численным интегрированием

P3	↑	F2	÷	P4	1	P8	ΠΠ	F7	1	ΠΠ	4
4	ΠΠ	4	2	БП	FXY	F8	3	÷	↑	F4	×
C/Π	БП	P0	×	↑	F8	+	P8	F2	1	—	P2
$x \neq 0$	P3	F3	↑	F4	—	P3	F3	x^2	↑	π	×
2	÷	cos	B/O								

При $x \gg 1$ целесообразно воспользоваться асимптотическими разложениями

$$\begin{aligned} C(x) &= \frac{1}{2} + \frac{\sin \frac{\pi x^2}{2}}{\pi x} \left(1 - \frac{3}{(\pi x^2)^2} + \frac{3 \cdot 7}{(\pi x^2)^4} - \dots \right) - \\ &\quad - \frac{\cos \frac{\pi x^2}{2}}{\pi^2 x^3} \left(1 - \frac{5}{(\pi x^2)^2} + \frac{5 \cdot 9}{(\pi x^2)^4} - \dots \right); \\ S(x) &= \frac{1}{2} - \frac{\cos \frac{\pi x^2}{2}}{\pi x} \left(1 - \frac{3}{(\pi x^2)^2} + \frac{3 \cdot 7}{(\pi x^2)^4} - \dots \right) - \\ &\quad - \frac{\sin \frac{\pi x^2}{2}}{\pi^2 x^3} \left(1 - \frac{5}{(\pi x^2)^2} + \frac{5 \cdot 9}{(\pi x^2)^4} - \dots \right). \end{aligned}$$

Вычисления по этим формулам при ограничении рядов, заключенных в скобки, двумя первыми членами реализовано программой 69.

Программа 69. Вычисление $C(x)$ и $S(x)$ по асимптотическим разложениям

↑	π	×	P2	×	P3	2	÷	e^{ix}	P4	XY	P5
F3	x^2	3	XY	÷	1	—	↑	F5	×	↑	F2
÷	P6	F3	x^2	5	XY	÷	1	—	↑	F4	×
↑	F3	÷	↑	F2	÷	2	$1/x$	+	↑	F6	—
C/Π	F4	/—/	↑	F5	/—/	P4	XY	P5	БП	P2	

При использовании программы 69 после ввода аргумента в регистр X нажимают клавиши В/О и С/П и на индикаторе высвечивается значение $C(x)$. Затем нажимают клавишу С/П, что приводит к высвечиванию значения $S(x)$. Погрешность результата при $x > 0,73$ не превышает $5/\pi^4 x^7$, что уже для $x = 4$ дает пять верных знаков после запятой. Правильность занесения в УЗП программы 69 можно проверить по контрольным значениям $C(4) = 0,4984198$ при времени счета около 10 с, $S(4) = 0,4205170$ при времени счета около 8 с.

Для вычисления коэффициентов многочленов Лежандра

$$P_n(x) = \frac{1 \cdot 3 \cdot 5 \dots (2n-1)}{n!} \left[x^n - \frac{n(n-1)}{2(2n-1)} x^{n-2} + \right. \\ \left. + \frac{n(n-1)(n-2)(n-3)}{2 \cdot 4 \cdot (2n-1)(2n-3)} x^{n-4} \dots \right],$$

многочленов Эрмита [2]

$$H_n(x) = x^n - \frac{n(-1)}{2} x^{n-2} + \frac{n(n-1)(n-2)(n-3)}{2 \cdot 4} x^{n-4} \dots,$$

многочленов Лагерра

$$L_n(x) = (-1)^n (n!)^2 \left[\frac{x^n}{(n!)^2} - \frac{x^{n-1}}{1! [(n-1)!]^2} + \frac{x^{n-2}}{2! [(n-2)!]^2} - \dots \right]$$

и многочленов Чебышева

$$T_n(x) = 2^{n-1} \left[x^n - \frac{n}{1! 2^2} x^{n-2} + \frac{n(n-3)}{2! 2^4} x^{n-4} - \dots \right]$$

можно воспользоваться программами 70—73 соответственно.

Программа 70. Вычисление коэффициентов многочленов Лежандра

$$\begin{array}{ccccccccc} P2 & 1 & P6 & P8 & + & P3 & 2 & P7 & \times & 3 & - & P5 \\ P4 & \uparrow & F8 & \times & \uparrow & F6 & \div & P8 & F6 & 1 & + & P6 \\ F4 & 2 & - & x < 0 & P2 & F8 & C/P & F3 & \cdot 1 & - & 1 & - \\ P3 & \times & \uparrow & F8 & \times & \uparrow & F7 & \div & \uparrow & F5 & \div & /- \\ P8 & F7 & 2 & + & P7 & F5 & 2 & - & P5 & БП & , \end{array}$$

Программа 71. Вычисление коэффициентов многочленов Эрмита

$$\begin{array}{ccccccccc} P2 & 1 & P3 & P8 & + & P4 & 2 & P5 & F8 & C/P & F4 & 1 \\ - & 1 & - & P4 & \times & \uparrow & F8 & \times & \uparrow & F5 & \div & /- \\ P8 & F5 & 2 & + & P5 & БП & РХҮ & & & & & \end{array}$$

Программа 72. Вычисление коэффициентов многочленов Лагерра

$$\begin{array}{ccccccccc} P2 & 0 & P3 & \pi & \times & \cos & P8 & C/P & F3 & 1 & + & P3 \\ F2 & - & x^2 & \uparrow & F8 & \times & \uparrow & F3 & \div & /- & БП & P1 \end{array}$$

Программа 73. Вычисление коэффициентов многочленов Чебышева

$$\begin{array}{ccccccccc} P2 & 2 & P4 & P8 & - & P3 & F4 & 2 & \times & P4 & F3 & 1 \\ P5 & - & P3 & x = 0 & P1 & F4 & C/P & /- & \uparrow & F2 & \times & 4 \\ \div & \uparrow & F5 & \times & C/P & F8 & \div & P4 & F8 & 1 & P5 & + \\ P8 & P7 & 2 & - & P6 & F7 & 1 & + & P7 & F2 & - & \uparrow \\ F5 & \times & P5 & F6 & 1 & - & P6 & x = 0 & БП & F4 & БП & \div \end{array}$$

Перед пуском каждой из этих программ заданное значение n заносят в регистр x , после чего нажимают клавиши В/О и С/П, что приводит к высвечиванию коэффициента при x^n . Для вычисления последующих коэффициентов нажимают только клавишу С/П до тех пор, пока высвечиваемое на индикаторе число не станет равным нулю. Проверку правильности занесения в УЗП программ 70—73 можно выполнить по контрольным результатам вычислений

$$P_7(x) = 26,8125x^7 - 43,3125x^5 + 19,6875x^3 - 2,1875x;$$

$$H_{10}(x) = x^{10} - 45x^8 + 630x^6 - 3150x^4 + 4725x^2 - 945;$$

$$L_7(x) = -x^7 + 49x^6 - 882x^5 + 7350x^4 - 29400x^3 +$$

$$+ 52920x^2 - 35280x + 5040;$$

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

Коэффициенты многочленов Эрмита, Лагерра и Чебышева являются целыми числами и результаты вычислений должны быть при необходимости соответственно округлены.

Для вычисления корней многочлена Чебышева и их экстремальных значений удобно использовать представление

$$T_n(x) = \begin{cases} \cos(n \arccos x) & \text{при } |x| < 1; \\ \operatorname{ch}(n \operatorname{arch} x) & \text{при } |x| > 1. \end{cases}$$

В этом случае равные единице максимумы многочлена соответствуют значениям аргумента $x = \cos(2\pi k/n)$, вычисляемым по программе 74, равные -1 минимумы соответствуют значениям аргумента $x = \cos \pi (2k + 1)/n$, вычисляемым по программе 75, а корни многочлена $x = \cos \pi(2k + 1)/2n$ (где k — целое число, удовлетворяющее условию $|k| < n$) вычисляются по программе 76.

Программа 74. Вычисление ординат максимумов многочлена Чебышева

P2	1	+	2	÷	P3	π	2	\times	C/P	F5	↑	F2	÷
P4	0	P5	F4	↑	F5	\times	cos	C/P	1	+			
P5	↑	F3	—	$x \geqslant 0$	2	0	0	C/P					

Программа 75. Вычисление ординат минимумов многочлена Чебышева

P2	π	↑	F2	÷	P3	1	P4	F3	↑	F4	\times
cos	C/P	F4	2	+	P4	F2	—	1	+	/—/	$x < 0$
PXY	0	0	C/P								

Программа 76. Вычисление корней многочленов Чебышева

P2	π	2	÷	↑	F2	÷	P3	0	P4	F4	2
\times	1	+	↑	F3	\times	cos	C/P	F4	1	+	P4
↑	F2	—	$x \geqslant 0$	FXY	0	0	C/P				

При работе с программами 74—76 значение n вводят в индикаторный регистр и после нажатия клавиши В/О и С/П вычисляют первый результат. Для вычисления последующих результатов нажимают только клавишу С/П, причем после вычисления последнего результата высвечиваются цифры 00, а сам результат заносится в регистр 3.

8. ЦИФРОВОЕ МОДЕЛИРОВАНИЕ

Обычный подход к математическому моделированию физических объектов заключается в составлении и решении уравнений, связывающих реакции $y(x)$ на воздействия $x(t)$, задаваемые в некотором интервале изменения t графическими, аналитическими или табличными моделями.

Более гибкий подход связан с использованием аналоговых вычислительных машин (АВМ). Соединяя надлежащим образом отдельные блоки АВМ, выполняющие определенные математические операции (сложение, дифференцирование и т. д.) над электрическими сигналами, составляют аналоговую модель исследуемого объекта. Подавая на вход такой модели электрический сигнал $x(t)$, аналогичный по форме заданному воздействию, на выходе модели получают сигнал, отображающий реакцию. Преимущества аналогового моделирования в реальном масштабе времени связаны с относительной простотой его реализации и возможностью исследования влияния воздействий произвольной формы без их предварительного математического описания. Недостатки этого способа моделирования в основном определяются большой погрешностью вычислений на АВМ.

Подход, близкий к аналоговому моделированию, возможен и при использовании цифровых ЭВМ. В этом случае моделирующий воздействие $x(t)$ электрический сигнал предварительно преобразуется аналогово-цифровым преобразователем (АЦП) в последовательность дискретных значений $x_i = x(t_0 + i\theta)$, где $\theta = \Delta t$ — шаг дискретизации. Свойства исследуемого объекта моделируют специальной программой (цифровой моделью), по которой над каждым значением x_i производятся математические операции, приводящие к ис- комому значению реакции $y(x_i)$.

Цифровые модели или программы, даже не имеющие аналоговых прототипов, используются также для обработки текущей информации, носителем которой является входной электрический сигнал $x(t)$, в реальном масштабе времени. Примером подобного устройства может служить цифровой фильтр, пропускающий спектральные составляющие входного сигнала в соответствии с заданной частотной характеристикой. Цифровым фильтром является любая цифровая модель инерционного (следовательно, частотно-зависимого) канала причинно-следственной связи.

Основное преимущество цифровых моделей и устройств цифровой обработки информации связано с высокой точностью и надеж-

ностью вычислений на цифровых ЭВМ. При дискретизации непрерывных входных сигналов точность моделирования аналоговых прототипов повышается при уменьшении шага дискретизации, но при этом повышаются требования к быстродействию устройств цифровой обработки информации в реальном масштабе времени.

Ручной ввод исходных данных не позволяет непосредственно использовать программируемые микрокалькуляторы для цифрового моделирования в реальном масштабе времени. Однако при предварительном представлении заданного воздействия его табличной моделью с постоянным шагом θ возможно цифровое моделирование при ручном вводе последовательности значений воздействия x_i и регистрации выводимых на индикатор значений реакций y_i . Во

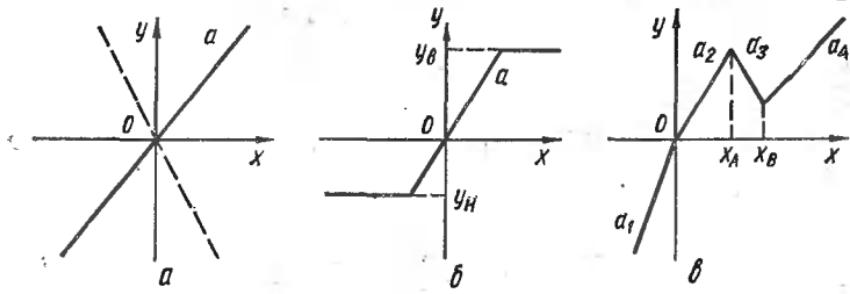


Рис. 34. Графические модели передаточных функций безынерционных каналов

многих случаях подобное цифровое моделирование упрощает расчеты и позволяет получить результаты, практически недостижимые при обычном подходе к моделированию [26].

Цифровая модель линейного безынерционного канала связи, описываемого уравнением (3), на используемом входном языке представляет собой программу

$\uparrow \text{F2} \times \text{С/П БП Р0}$

с исходными данными $a = \text{P2}$, $x_i = \text{PX}$ и результатом $\text{PX} = y_i$. Подобная модель графически отображается характеристикой $y(x)$ в виде прямой с наклоном a , проходящей в четных квадрантах при $a < 0$ (штриховая линия на рис. 34, a) и в нечетных при $a > 0$ (сплошная линия на рис. 34, a) через начало координат. Если содержимое регистра 2 изменять перед каждым выполнением приведенной программы по некоторому закону (в частности — по закону распределения случайных чисел для каналов с нестабильными параметрами), то эта программа будет моделировать параметрический канал связи. Если под содержимым регистра 2, изменяемым перед каждым вводом x_i , подразумевать некоторое стороннее воздействие z_i , то программа будет цифровой моделью устройства, перемножающего сигналы x_i и z_i .

Безынерционный канал с петлей обратной связи, описываемый уравнением (7), отображается цифровой моделью в виде программы 77.

Программа 77. Цифровая модель безынерционного канала с петлей обратной связи

$$\frac{\uparrow}{\text{БП}} \quad \begin{matrix} F2 \\ x \geqslant 0 \end{matrix} \quad \times \quad \begin{matrix} \uparrow \\ P3 \end{matrix} \quad \begin{matrix} F5 \\ \text{БП} \end{matrix} \quad \times \quad \begin{matrix} P6 \\ \div \end{matrix} \quad \begin{matrix} \uparrow \\ F3 \end{matrix} \quad \begin{matrix} F6 \\ \uparrow \end{matrix} \quad \begin{matrix} F4 \\ XY \end{matrix} \quad \times \quad \begin{matrix} 1 \\ \div \end{matrix} \quad \begin{matrix} C/P \\ \end{matrix}$$

$$\frac{\text{P0}}{\quad}$$

Исходные данные, соответствующие входящим в уравнение (7) параметрам, перед пуском программы размещаются следующим образом: $\theta = P_2$, $\mu = P_3$, $\beta = P_4$, $y_c = P_5$. Здесь y_c — значение реакции, соответствующее переходу анализируемого канала в новое устойчивое состояние при самовозбуждении ($\beta\mu \geqslant 1$) и определяемое нелинейными свойствами моделируемого объекта (например, переходом электронной лампы или транзистора в режим насыщения). При $\beta\mu < 1$ цифровая модель 77 связывает воздействие и реакцию в соответствии с формулой (7). Значения x_i вводят в регистр X с нажатием клавиши В/О и С/П после ввода x_0 и только клавиши С/П после ввода последующих значений. Изменяя содержимое регистров 2—5, можно моделировать изменение параметров анализируемого канала передачи воздействия во времени.

Более общая модель петли обратной связи (или, например, многократного отражения волн от концов отрезка линии с распределенными параметрами) формируется в виде замкнутого цикла с помощью операторов перехода. Однако такая модель более сложна и требует больших затрат времени на вычисления, в связи с чем ее целесообразно использовать лишь в случае необходимости — например, при инерционности передач.

Обратную связь в автономных системах можно моделировать программами, в которых (после вычисления реакции на некоторое начальное воздействие x_0) в качестве воздействий x_i используется сохраняемое в индикаторном регистре ранее вычисленное значение y_{i-1} .

Нелинейные каналы связи с ограничением выходного сигнала моделируются с помощью условных операторов, контролирующих численное значение переменной реакции. Например, нелинейный безынерционный канал с двусторонним ограничением (рис. 34, б) описывается цифровой моделью в виде программы 78 ($a = P_5$, $y_a = P_6$, $y_b = P_7$, $x_i = PX$, $PX = y_i$).

Программа 78. Цифровая модель двустороннего ограничителя

$$\frac{\uparrow}{x \geqslant 0} \quad \begin{matrix} F5 \\ F \times \end{matrix} \quad \times \quad \begin{matrix} \uparrow \\ XY \end{matrix} \quad \begin{matrix} F7 \\ P8 \end{matrix} \quad - \quad \begin{matrix} x < 0 \\ \text{БП} \end{matrix} \quad \begin{matrix} F \times \\ C/P \end{matrix} \quad \begin{matrix} F6 \\ P0 \end{matrix} \quad -$$

Подобная модель приближенно отображает, в частности, свойства активных усилительных компонентов электронных цепей (ламп или транзисторов) в режиме сильного сигнала.

С помощью условных операторов моделируются также нелинейности типа упора в механических системах, при которых увеличение

входного воздействия больше некоторого порогового значения не вызывает увеличения переменной реакции.

При сложном характере нелинейной зависимости $y(x)$ безынерционного канала часто используют кусочно-линейную аппроксимацию его статической характеристики, моделируемой с помощью условных операторов, определяющих принадлежность значения входного сигнала определенному участку аппроксимированной характеристики. Примером может служить цифровая модель канала с N -образной статической характеристикой (например, вольт-амперной характеристикой туннельного диода) в виде программы 79

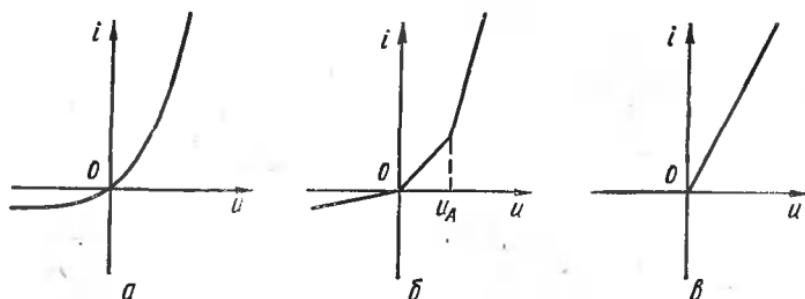


Рис. 35. Аппроксимация вольт-амперных характеристик диодов

($a_1 = P2$, $a_2 = P3$, $a_3 = P4$, $a_4 = P5$, $x_A = P6$, $x_B = P7$, $x_i = = PX$, $PX = y_i$), моделирующей кусочно-линейную аппроксимацию подобной характеристики (рис. 34, б).

Программа 79. Цифровая модель канала с N -образной характеристикой

\uparrow	F2	\times	$x \geq 0$	7	F3	\times	\rightarrow	F6	$-$	$x \geq 0$	PCx
F4	\times	\rightarrow	F7	$-$	$x \geq 0$	5	F5	\times	\rightarrow	F4	\uparrow
F5	$-$	\uparrow	F7	\times	\uparrow	\leftarrow	$+$	\rightarrow	F3	\uparrow	F4
$-$	\uparrow	F6	\times	\uparrow	\leftarrow	$+$	\rightarrow	\leftarrow	C/P	BП	P0

Программа 79 автоматически определяет интервал, которому принадлежит введенное значение x_i , и вычисляет значение реакции по формулам

$$y_i = \begin{cases} a_1 x_i, & \text{если } x_i < 0; \\ a_2 x_i, & \text{если } 0 \leq x_i < x_A; \\ a_3 x_i + (a_2 - a_3) x_A, & \text{если } x_A \leq x_i < x_B; \\ a_4 x_i + (a_2 - a_3) x_A + (a_3 - a_4) x_B, & \text{если } x_B \leq x_i. \end{cases}$$

Иногда для цифровой модели целесообразно использовать аналитическое выражение, связывающее реакцию и воздействие. Примером может служить аналитическое выражение статической характеристики (рис. 35, а) полупроводникового диода

$$i = A (e^{\Lambda u} - 1),$$

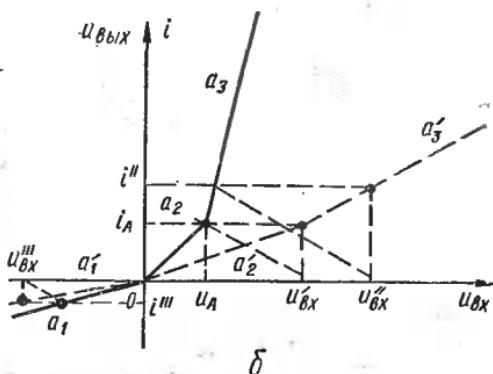
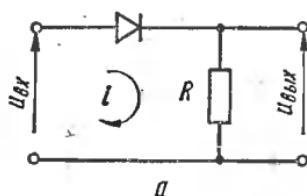
по которому легко составить цифровую модель в виде программы 80 ($A = P2$, $\Lambda = P3$, $u_i = PX$, $PX = i_i$) для вычисления значений тока i через диод по значениям напряжения u на диоде.

Программа 80. Цифровая модель статической характеристики диода

↑ F3 × e^x 1 — ↑ F2 × С/Π БΠ Р0

В большинстве инженерных задач удобна кусочно-линейная аппроксимация статической характеристики диода (рис. 35, б), отображаемая цифровой моделью в виде программы 81 ($a_1 = P2$, $a_2 = -P3$, $a_3 = P4$, $a_4 = P5$, $u_i = PX$, $PX = i_i$).

Рис. 36. Нагруженный диод:
а — схема замещения; б — аппроксимация передаточной характеристики



Программа 81. Цифровая модель кусочно-линейной статической характеристики диода

↑ F2 × $x \geq 0$ F4 F3 × → F5 — $x \geq 0$ P4
 ↓ F4 × → F3 ↑ F5 × ↑ ← + →
 ← C/Π БΠ P0

Вместе с этим во многих задачах достаточно описать полупроводниковый или вакуумный диод характеристикой идеального вентиля (рис. 35, б), представляемого простой цифровой моделью в виде программы 82 ($a = P2$, $u_i = PX$, $PX = i_i$).

Программа 82. Цифровая модель идеального вентиля

↑ $x < 0$ ↑ 0 ↑ F2 × C/Π БΠ Р0

Необходимо подчеркнуть, что выбор цифровой модели элемента некоторой системы определяется не только числом операторов, временем счета или точностью результатов, но и возможностью ее соединения с цифровыми моделями других элементов системы. Например, в схеме нагруженного диода (рис. 36, а) воздействие $u_{\text{вх}}$ связано с реакцией $u_{\text{вых}} = iR$ соотношением

$$u_{\text{вых}} = RA [e^{\Lambda(u_{\text{вх}} - u_{\text{вых}})} - 1].$$

Если взять за основу аналитическую модель, то для каждого вводимого значения $u_{\text{вх}}$ необходимо решать трансцендентное нелинейное уравнение, что связано с большими затратами времени. С помощью кусочно-линейной аппроксимации статической характеристики диода можно сформировать более простую цифровую модель без замкнутых циклов. Для этого на характеристике диода с кусочно-линейной аппроксимацией (рис. 36, б) из точки с координатами (i_A, u_A) проведем отрезок прямой под наклоном $R = \Delta u / \Delta i$ до пересечения с осью напряжений в точке $u'_{\text{вх}}$, а через точки $u''_{\text{вх}} > u'_{\text{вх}}$ и $u'''_{\text{вх}} < 0$ — параллельные этому отрезку прямые до пересечения с характеристикой диода. Соединив теперь точки с координатами $(u''_{\text{вх}}, i'')$, $(u'''_{\text{вх}}, i'')$ и $(u'_{\text{вх}}, i_A)$ отрезками прямых, получим кусочно-линейную характеристику нагруженного диода. Определив для этих отрезков наклоны a'_1 , a'_2 и a'_3 с учетом масштаба $u_{\text{вых}} = R_i$, можно использовать программу 81 при $u'_{\text{вх}} = P5$, $u_{\text{вх}_i} = PX$, $PX = u_{\text{вых}_i}$.

Следует добавить, что даже в линейных системах параметры входного и выходного сигналов могут быть связаны нелинейными зависимостями. Для примера рассмотрим связь между амплитудой и частотой входного напряжения и напряжением на выходе дифференциального частотного детектора в квазистационарном режиме [20], когда инерционностью можно пренебречь. Эта связь описывается формулой

$$U_{\text{вых}}(t) \approx K_a U_{\text{вх}}(t) \frac{\sqrt{1 + (0,5\eta + \xi)^2} - \sqrt{1 + (0,5\eta - \xi)^2}}{\sqrt{(1 + \xi^2 - \eta^2)^2 + 4\eta^2}},$$

где K_a — коэффициент передачи напряжения диодами; $u_{\text{вх}}(t)$ — амплитуда входного сигнала; ξ — обобщенная расстройка (нормированная частота); η — параметр связи между контурами.

При $K_a=1$ (передача диодов не зависит от амплитуды входного сигнала) это выражение моделируется программой 83, позволяющей вычислить реакцию ЧМ-детектора на воздействие входным сигналом, задаваясь соответственно в регистрах Y и X отсчетами амплитуды $u_{\text{вх}}(t)$ и нормированной частоты $\xi = 2(\omega - \omega_p)Q/\omega_p$, где ω_p — резонансная частота контуров, а Q — их добротность. Если амплитуда $U_{\text{вх}}(t)$ постоянна, то программу можно упростить, заменив первые четыре оператора одним Р3 и занеся в регистр 4 фиксированное значение $U_{\text{вх}}$ и 0,5 η в регистр 2.

Программа 83. Цифровая модель частотного детектора

XY	P4	XY	P3	x^2	1	+	4	\div	\uparrow	F2	x^2
XY	—	x^2	+	$\sqrt{ }$	\rightarrow	F3	\uparrow	F2	—	x^2	1
+	$\sqrt{ }$	\rightarrow	F3	\uparrow	F2	+	x^2	1	+	$\sqrt{ }$	\uparrow
\leftarrow	—	\uparrow	\leftarrow	\div	4	\div	\uparrow	F4	\times	C/P	BП
P0											

Более сложно цифровое моделирование инерционных каналов связи. Идеальный интегрирующий канал (интегратор) моделируется программой 40 для численного интегрирования табличных моделей. Идеальный дифференцирующий канал (дифференциатор) моделируется программой 84 ($\theta = P2$, $x_i = PX$, $PX = y_i$), реализующей вычисления по формуле

$$y_i = dx/dt = (x_i - x_{i-1})/\theta.$$

Программа 84. Цифровая модель дифференцирующего канала

$\uparrow F3 - XY \quad P3 \quad F2 \div C/P \quad B/P \quad P0$

Подобные модели можно использовать, в частности, при цифровом моделировании систем автоматического регулирования. Однако цифровое моделирование устройств обработки информации соединением интеграторов и дифференциаторов часто оказывается чрезмерно громоздким и приходится прибегать к более общим методам построения цифровых фильтров.

Теория цифровых фильтров основана на линейных разностных уравнениях, заменяющих для дискретных функций обыкновенные дифференциальные уравнения. Линейный непараметрический фильтр, описываемый разностным уравнением (77) с постоянными коэффициентами, называют рекурсивным. Если же выходной сигнал представляет собой взвешенную сумму M предыдущих значений только входной последовательности ($W_0 = 1$, $W_k = 0$), то фильтр называют нерекурсивным. Соединяя программы, моделирующие цифровые фильтры и безынерционные устройства, можно формировать цифровые модели, описывающие свойства достаточно сложных устройств передачи воздействия или обработки информации [26].

Согласно уравнению (77) при моделировании нерекурсивного фильтра необходимо хранить M предыдущих значений входной последовательности и $M + 1$ весовых коэффициентов V_k , определяющих импульсную характеристику фильтра. Так как еще один регистр памяти нужен для накопления суммы, определяющей выходной сигнал y_i , то при 15 регистрах (включая операционные) можно промоделировать нерекурсивный фильтр до шестого порядка включительно программой 85 ($V_0 = P2$, $V_1 = P3$, $V_2 = P4$, $V_3 = P5$, $V_4 = P6$, $V_5 = P7$, $V_6 = P8$, $x_i = PX$, $PX = y_i$).

Программа 85. Модель нерекурсивного фильтра

\rightarrow	\uparrow	F8	\times	\rightarrow	\uparrow	F7	PП	P5	F6	PП	P5
F5	PП	P5	F4	PП	P5	F3	PП	P5	F2	\times	\leftarrow
XY	\rightarrow	$+$	C/P	B/P	P0	\times	\leftarrow	XY	\rightarrow	$+$	\rightarrow
\uparrow	B/O										

При использовании этой программы после ввода x_0 нажимают клавиши В/О и С/П, после ввода остальных отсчетов входного сигнала — только клавишу С/П. При надлежащем выборе коэффициентов V_k программа 85 тем точнее моделирует свойства аналогового прототипа цифрового фильтра, чем меньше шаг дискретизации θ .

При работе с моделями цифровых фильтров импульсную характеристику (описывающую реакцию на единичный импульс) вычисляют при вводе последовательности $x_i = 1, 0, 0, \dots$, а переходную характеристику (описывающую реакцию на единичное включение) — при вводе последовательности $x_i = 1, 1, 1, \dots$. Поэтому при использовании программы 85 или других моделей цифровых фильтров вычисление импульсной или переходной характеристики можно упростить, вводя между операторами С/П и БП оператор занесения в регистр X нуля (0 или C_x) или единицы (1) соответственно. В этом случае после ввода $x_0 = 1$ с нажатием клавишей В/О и С/П в дальнейшем нажимают только клавишу С/П без ввода входного сигнала, который формируется автоматически.

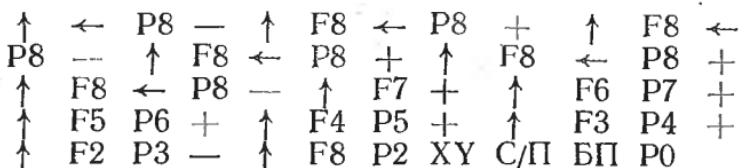
Весовые коэффициенты нерекурсивного фильтра можно определить по отсчетам импульсной характеристики аналогового прототипа. Например; при цифровом моделировании низкочастотного фильтра первого порядка с импульсной характеристикой

$$V_k = \frac{\theta}{\tau} e^{-k\theta/\tau} \mathbf{1}(k)$$

при $\theta/\tau = 0,5$ для программы 85 в качестве исходных данных можно принять $V_0 = 0,5$; $V_1 = 0,3032653$; $V_2 = 0,1839397$; $V_3 = 0,111565$; $V_4 = 0,06766767$; $V_5 = 0,04104252$; $V_6 = 0,02489355$. В этом случае при вводе $x_i = 1$ переходная характеристика отображается вычисляемой по программе 85 последовательностью $y_0 = 0,5$; $y_1 = 0,8032653$; $y_2 = 0,9872051$; $y_3 = 1,098770$; $y_4 = 1,166437$ и т. д. Время вычисления каждого отсчета 10—15 с. При меньшем значении θ/τ и большем M получаются меньшие различия характеристик цифрового фильтра и его аналогового прототипа..

Если коэффициенты V_k повторяются или являются целыми числами, то часть из них можно записать непосредственно в программу, увеличивая порядок M фильтра. Пусть, например, необходимо выделить из смеси помехи и кодовой последовательности Баркера [4] эту последовательность. В этом случае можно воспользоваться моделью нерекурсивного фильтра с коэффициентами $V_k = \pm 1$, следующими в порядке, обратном выделяемой последовательности. Так, для кода Баркера 1,1,1,1,1,—1,—1,1,1,—1,1,—1,1 порядка 13 коэффициенты согласованного нерекурсивного фильтра $V_0 = V_2 = V_4 = V_5 = V_8 = V_9 = V_{10} = V_{11} = V_{12} = 1$; $V_1 = V_3 = V_6 = V_7 = -1$. По программе 86 ($x_i = PX$, $PX = y_i$), реализующей подобный фильтр, каждый отсчет реакции y_i вычисляется за 18—19 с.

Программа 86. Модель нерекурсивного согласованного фильтра



При вводе исходной последовательности в виде кода Баркера на выходе согласованного фильтра, моделируемого программой 86, будет вычисляться последовательность отсчетов реакции $y_t = 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 13$. Контрольные значения реакции такого фильтра на смесь исходной кодовой последовательности и «нормального шума» с $\bar{\sigma} = 1$ и $\bar{m}_1 = 0$, приведены в табл. 50. Как следует из дан-

Таблица 50

t	x_t	y_t	i	x_i	y_i
0	0,8	0,8	11	-0,15	-2,35
1	-0,25	-1,05	12	0,05	4
2	1,6	2,65	13	-0,85	-1,15
3	-1,25	-3,9	14	-3,8	!
4	-0,05	3,85	15	2,5	4,5
5	0,2	-2,05	16	2,2	-7,85
6	-0,7	0,85	17	-0,6	1,5
7	0,1	0,85	18	0,25	0,85
8	0,7	-2,15	19	-1,2	0,65
9	1,55	2	20	1,65	18,4
10	3,2	5,2	21	-1,75	1,85

ных таблицы, фильтр выделяет кодированный сигнал из его смеси с шумом.

Для моделирования цифровых рекурсивных фильтров прежде всего необходимо по заданным требованиям определить их характеристики. Известны различные методы синтеза цифровых фильтров [6], но здесь остановимся кратко лишь на преобразованиях, позволяющих читателю, не знакомому с теорией цифровой фильтрации, формально рассчитать цифровой фильтр по заданному аналоговому прототипу.

В соответствии с соотношениями (78) по передаточной функции (66) можно составить разностное уравнение, с помощью которого можно непосредственно по отсчетам входного воздействия x_t вычислить отсчеты y_t реакции. Однако преобразование (78а) нелинейно относительно частоты, что приводит к деформации частотной оси при переходе от аналогового прототипа к цифровому фильтру. Это явление можно учесть, используя соотношение

$$\omega_A = \operatorname{tg}(\omega_d \theta / 2), \quad (144)$$

где ω_A и ω_n — соответственно характеристические частоты (например, частота резонанса или частота среза) аналогового прототипа и цифрового фильтра. Шаг (период) дискретизации θ определяет согласно теореме Котельникова [4] верхнюю частоту рабочего диапазона цифрового фильтра $f_v < \frac{1}{2}\theta$. Например, при $\theta = 1$ мкс рабочий диапазон фильтра ограничен частотой $f < 500$ кГц.

Рассмотрим примеры составления программ, моделирующих цифровые фильтры различного порядка.

Аналоговый фильтр нижних частот первого порядка описывается передаточной функцией вида

$$F(p) = 1/(1 + p\tau). \quad (145a)$$

В соответствии с формулами (78) находим

$$F(z) = \frac{1}{1 + [(z-1)/(z+1)]\tau} = \frac{z+1}{(1+\tau)z + (1-\tau)} = \frac{1+z^{-1}}{(1+\tau) + (1-\tau)z^{-1}}.$$

По этому выражению получаем разностное уравнение

$$y(i) = [x(i) + x(i-1) - (1-\tau)y(i-1)]/(1+\tau), \quad (145b)$$

которое несложно моделировать программой 87 ($\tau = P4$, $x_t = PX$, $PX = y_i$).

Программа 87. Модель фильтра низких частот первого порядка

$$\begin{array}{ccccccccc} \rightarrow & F4 & 1 & - & \uparrow & F3 & \times & \uparrow & F2 \\ P2 & + & \rightarrow & F4 & 1 & + & \uparrow & \leftarrow & XY \\ \text{БП} & P0 & & & & & & \div & P3 \\ & & & & & & & & C/P \end{array}$$

В процессе вычислений по программе 87 используются только три регистра ЗУПВ. Время вычислений каждого отсчета реакции около 4 с.

Пример 52. Промоделировать цифровой фильтр нижних частот первого порядка при $\theta = 100$ мкс, затухании 0 дБ на частоте $f = 0$ и 3 дБ на частоте $f_c = 1$ кГц.

Аналоговый фильтр с передаточной функцией (145a) должен иметь затухание 3 дБ на определяемой формулой (144) частоте среза $\omega_A = \operatorname{tg} \frac{\omega_n \theta}{2} = 0,3249196$.

Из соотношения для уровня 3 дБ

$$|K(j\omega_A)| = \sqrt{\frac{1}{1 + \omega_A^2 \tau^2}} = \frac{1}{\sqrt{2}}$$

вычисляем $\tau = 3,077684$.

Проверку правильности занесения в УЗП программы 87 с вычисленным значением τ выполним по переходной характеристике моделируемого фильтра (при вводе $x_t = 1$) $y_t = 0,2452372; 0,6154291; 0,8040513; 0,9001591; 0,9491285; 0,9740796; 0,9867929 \dots$

Аналоговый полосовой фильтр второго порядка (резонатор) описывается для обычно используемого соединения его элементов функцией входного (передаточного) сопротивления

$$F(p) = ap/(p^2 + b_1 p + b_0). \quad (146a)$$

После билинейного преобразования (78а) получим

$$F(z) = \frac{a(1-z^{-2})}{(1+b_0+b_1)+2(b_0-1)z^{-1}+(1+b_0-b_1)z^{-2}},$$

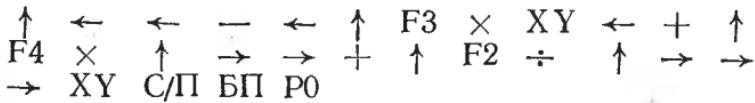
откуда разностное уравнение

$$y(i) = [x(i) - x(i-2) - W_1 y(i-1) - W_2 y(i-2)]/W_0, \quad (146b)$$

где $W_1 = 2(b_0 - 1)/a$; $W_2 = (1 + b_0 - b_1)/a$; $W_0 = (1 + b_0 + b_1)/a$.

Полученное уравнение реализуется программой 88 ($W_0 = P2$, $W_1/W_2 = P3$, $-W_2 = P4$, $x_i = PX$, $PX = y_i$).

Программа 88. Модель цифрового резонатора



Пример 53. Построить цифровую модель аналогового резонатора, образованного параллельным соединением эквивалентных емкости C , индуктивности L и проводимости $C = 1/R$ при $f_p = 10$ кГц, $Q = \omega_p CR = 10$, $R = 1$ кОм, $\theta = 10$ мкс.

Входное сопротивление аналогового прототипа фильтра описывается выражением

$$F(p) = \frac{p\omega_p dR'}{(p^2 + \omega_p dp + \omega_p^2)},$$

где $\omega_p = 1/\sqrt{LC}$; $d = 1/Q$; $R' = R \cdot 10^{-3}$ для более удобного выражения входного тока в миллиамперах.

Вычисляем резонансную частоту аналогового прототипа с учетом частотных искажений при переходе к цифровому фильтру $\omega_A = \operatorname{tg} \pi \cdot 10^{-1} = 0,325$ и в соответствии с записью разностного уравнения (77) в виде уравнения (146б) вычисляем $W_0 = (1 + \omega_p^2 + \omega_p d)/(d\omega_p R') = 35,01923$; $W_1 = 2(\omega_p^2 - 1)/(d\omega_p R') = -55,03846$; $W_2 = (1 + \omega_p^2 - \omega_p d)/(d\omega_p R') = 33,01923$.

Вводя в УЗП программу 88 и вычисленные значения коэффициентов W_0 , W_1 и W_2 в соответствующие регистры памяти, получаем цифровую модель, описывающую связь между воздействием x_i и реакцией y_i моделируемого фильтра.

Аналоговый фильтр низких частот с максимально плоской характеристикой (фильтр Баттервортса) описывается функцией

$$F(p) = \frac{1}{(1+p\tau)(1+p\tau+p^2\tau^2)} = \frac{1}{1+2p\tau+2p^2\tau^2+p^3\tau^3}. \quad (147)$$

Переходя к z -функции цифрового фильтра

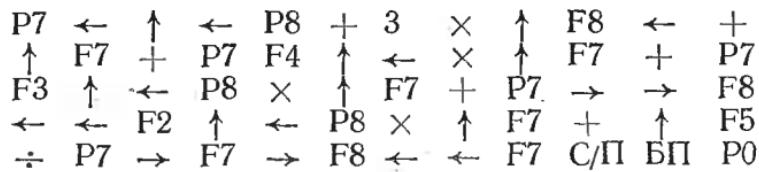
$$F(z) = \frac{1+3z^{-1}+3z^{-2}+z^{-3}}{(1+2\tau+2\tau^2+\tau^3)+(3+2\tau-2\tau^2-3\tau^3)z^{-1}+(3-2\tau-2\tau^2+3\tau^3)z^{-2}+(1-2\tau+2\tau^2-\tau^3)z^{-3}},$$

находим разностное уравнение

$$y(i) = \frac{[x(i) + 3x(i-1) + 3x(i-2) + x(i-3) - W_1y(i-1) - W_2y(i-2) - W_3y(i-3)]}{W_0}.$$

Это уравнение моделируется программой 89 ($-W_1 = P2, -W_2 = P3, -W_3 = P4, W_0 = P5, x_i = PX, PX = y_i$). Время обработки одного отсчета в микрокалькуляторе около 12 с.

Программа 89. Модель максимально плоского фильтра нижних частот третьего порядка



Пример 54. Промоделировать фильтр нижних частот третьего порядка с максимально плоской характеристикой при затухании 3 дБ на частоте 1 кГц и $\theta = 100$ мкс.

Частота цифрового фильтра 1 кГц соответствует частоте аналогового фильтра $\omega_A = \operatorname{tg}\pi \cdot 10^{-1} = 0,325$. По амплитудно-частотной характеристике фильтра на уровне 3 дБ

$$|K(j\omega)| = \frac{1}{\sqrt{1 + \omega_A^6 \tau^6}} = \frac{1}{\sqrt[6]{2}}$$

находим $\tau = 3,077684$ и вычисляем коэффициенты разностного уравнения $W_0 = 55,25191; W_1 = -97,24569; W_2 = 65,35713; W_3 = -15,36334$. Вводя вычисленные исходные данные и программу 89 в микрокалькулятор, получаем цифровую модель заданного фильтра третьего порядка.

Импульсную характеристику фильтра (или его реакцию на произвольное воздействие, описанную дробно-рациональной функцией комплексной частоты) можно вычислить по коэффициентам ряда (одностороннего z -преобразования)

$$F(z) = \frac{a_0 z^m + a_1 z^{m-1} + \dots + a_m}{b_0 z^n + b_1 z^{n-1} + \dots + b_n} = \sum_{i=0}^{\infty} f_i z^{-i}, \quad (148)$$

являющимися отсчетами оригинала функции y_i в моменты времени $i\theta$. В этом случае на используемом входном языке возможно моделирование произвольных фильтров до шестого порядка включительно ($m+n \leq 12$). Автоматическое разложение функции $F(z)$ в ряд (148) делением числителя на знаменатель выполняется по программе 90 ($b_0 = P2, b_1 = P3, b_2 = P4, b_3 = P5, b_4 = P6, b_5 = P7, b_6 = P8, a_1 = C1, a_2 = C2, a_3 = C3, a_4 = C4, a_5 = C5, a_6 = C6, a_0 = PX$).

Программа 90. Модель импульсной характеристики фильтра шестого порядка

$$\begin{array}{ccccccccc}
 \uparrow & F_2 & \div & \uparrow & F_3 & \Pi \Pi & F, & F_4 & \Pi \Pi \\
 F, & F_6 & \Pi \Pi & F, & F_7 & \Pi \Pi & F, & F_8 & \Pi \Pi \\
 C_x & \leftarrow & B \Pi & P_0 & \times & X Y & \leftarrow & X Y & - \\
 \leftarrow & B / O & & & & & & & \uparrow \rightarrow X Y \\
 & & & & & & & & C / \Pi
 \end{array}$$

При вычислении f_0 по этой программе нажимают клавиши B/O и C/P, при вычислении остальных коэффициентов — только клавишу C/P. Время вычисления каждого коэффициента около 18 с.

В качестве примера вычислим импульсную характеристику фильтра Чебышева шестого порядка [10], описываемую функцией

$$\begin{aligned}
 F(p) = & 1/[1 + 0,15178p + \\
 & + 0,97753p^2)(1 + 0,71902p + \\
 & + 1,69499p^2)(1 + 3,69181p + \\
 & + 6,37095p^2)].
 \end{aligned}$$

После перехода к z -функции в соответствии с формулами (78) и (144) при $\theta = 1$ мкс и $f_c = 100$ Гц находим

$$\begin{aligned}
 F(z) = & (z^6 + 6z^5 + 15z^4 + 20z^3 + 15z^2 + 6z + 1)/(15027,29z^6 - \\
 & - 71156,35z^5 + 147901,3z^4 - 171643,7z^3 + \\
 & + 116913,7z^2 - 44243,08z + 7264,616).
 \end{aligned}$$

Вводя коэффициенты полученной z -функции в регистры памяти микрокалькулятора, по программе 90 вычисляем отсчеты импульсной характеристики (рис. 37), практически полностью совпадающие с результатами расчета импульсной характеристики аналитическим методом (сплошная линия на рис. 37).

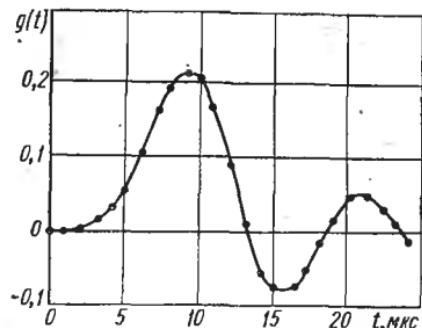


Рис. 37. Импульсная характеристика фильтра Чебышева шестого порядка

Глава 4

АВТОМАТИЗАЦИЯ РЕШЕНИЯ ТИПОВЫХ ЗАДАЧ

1. РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОЙ АЛГЕБРЫ

Решение многих инженерных задач связано с операциями линейной алгебры над элементами прямоугольных таблиц (матриц), обеспечивающими, в частности, вычисление неизвестных переменных в системе линейных уравнений. Для полной автоматизации подобных задач необходимо хранить в памяти микрокалькулятора все элементы матриц, над которыми выполняются операции. Поэтому допустимые размеры таких матриц при полностью автоматизированных вычислениях ограничены емкостью запоминающих устройств и относительно малы. Расширение этих размеров возможно с переходом в полуавтоматический режим решения задачи по частям, при котором промежуточные результаты вычислений записываются в вычислительный бланк, выполняющий функции внешнего запоминающего устройства.

В ряде задач возникает необходимость в сложении и умножении матриц. Для сложения двух матриц в обычном режиме использования микрокалькулятора требуется меньшее число нажатий клавиш (включая занесение программы в УЗП), чем в программируемом режиме. Поэтому программу 91 для сложения матриц (или любых упорядоченных множеств чисел), содержащих до шести элементов включительно, целесообразно использовать лишь при сложении нескольких однотипных матриц.

Программа 91. Сложение матриц, содержащих до шести элементов

F3	↑	→	+	P3	F6	↑	→	+	P6	F5	↑
→	+	P5	F4	↑	→	+	P4	F8	↑	→	+
P8	F7	↑	→	+	P7	0	0	C/П	БП	P0	

Перед пуском программы (для первого пуска нажимают клавиши В/О и С/П) элементы двух слагаемых матриц вводят в регистры памяти соответственно C1 и P7, C2 и P8, C3 и P4, C4 и P5, C5 и P6, C6 и P3 (если число суммируемых элементов меньше шести, то в свободные регистры заносятся нули), причем в стек исходные дан-

ные удобно вводить против часовой стрелки. После выполнения программы на индикаторе высвечиваются цифры 00, а в регистрах ЗУПВ исходные данные замещаются соответствующими результатами. Для суммирования полученной матрицы со следующей достаточно ввести элементы последней в стек и повторить выполнение программы.

Например, для сложения матриц

$$\begin{bmatrix} 2,1 & 3,5 \\ 4 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 6,7 & -5,1 \\ 1 & 21 \\ -3 & 2 \end{bmatrix} + \begin{bmatrix} 0,3 & -11 \\ -2,4 & -10 \\ 2,2 & 4 \end{bmatrix} = \begin{bmatrix} 9,1 & -12,6 \\ 2,6 & 11 \\ 0,2 & 6 \end{bmatrix}$$

после занесения программы 91 в УЗП следует ввести в регистры памяти элементы двух матриц (например, 2,1 = P7; 3,5 = P8; 4 = P4; 0 = P5; 1 = P6; 0 = P3; 6,7 = C1; -5,1 = C2; 1 = C3; 21 = C4; -3 = C5; 2 = C6). После выполнения программы и высвечивания цифр 00 результаты сложения хранятся в ЗУПВ (P7 = 8,8; P8 = -1,6; P4 = 5; P5 = 21; P6 = -2; P3 = 2). Элементы третьей матрицы вводят в стек (0,3 = C1; -11 = C2; -2,4 = C3; -10 = C4; 2,2 = C5; 4 = C6) и после выполнения программы в соответствующих регистрах ЗУПВ оказываются занесенными результаты сложения всех трех матриц.

Если суммируемые матрицы (или другие упорядоченные множества чисел) имеют более шести элементов, то сложение с помощью программы 91 можно выполнять по частям из шести или менее элементов.

Произведение матрицы A размера $m \times n$ на матрицу B размера $n \times p$ равно прямоугольной матрице $C = AB$ размера $m \times p$ с элементами

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}; \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, p.$$

Произведение квадратных матриц второго порядка

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

можно вычислить по программе 92 ($a_{11} = C4$, $a_{12} = C3$, $a_{21} = C2$, $a_{22} = C1$, $b_{11} = P7$, $b_{12} = P8$, $b_{21} = P4$, $b_{22} = P5$). После выполнения программы (время счета около 10 с) на индикаторе высвечиваются цифры 7845, указывающие номера регистров ЗУПВ, в которых хранится результат умножения ($P7 = c_{11}$, $P8 = c_{12}$, $P4 = c_{21}$, $P5 = c_{22}$). При этом стек с элементами первой матрицы автоматически устанавливается в исходное положение.

Программа 92. Умножение матриц второго порядка

F4	P3	F7	P6	\leftarrow	ПП	6	\rightarrow	\rightarrow	ПП	6	P7
\rightarrow	\rightarrow	P4	\leftarrow	\leftarrow	F5	P3	F8	P6	\leftarrow	ПП	6
\rightarrow	\rightarrow	ПП	6	P8	\rightarrow	\rightarrow	P5	\leftarrow	\leftarrow	7	8
4	5	C/P	\uparrow	F3	\times	P2	XY	\leftarrow	\uparrow	\leftarrow	\leftarrow
\leftarrow	F6	\times	\uparrow	F2	$+$	B/O					

При использовании программы 92 для вычисления произведения нескольких матриц второго порядка после вычисления очередного произведения двух матриц в стек вводят предыдущую (записанную слева) матрицу и по программе вычисляется произведение этой матрицы на ранее полученную матрицу-произведение. Для каждого умножения нажимают клавиши В/О и С/П. Например, для вычисления произведения матриц

$$\begin{bmatrix} 5 & -2 \\ -3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 8,2 & 2 \\ 1 & 8,1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 24,6 & 6 \\ 8,2 & 2 \end{bmatrix}$$

в регистры памяти следует ввести элементы двух крайних матриц: $8,2 = C_4$; $2 = C_3$; $1 = C_2$; $8,2 = C_1$; $1 = P_7$; $0 = P_8$; $0 = P_4$; $1 = P_5$. После выполнения программы ($P_7 = 8,2$; $P_8 = 2$; $P_4 = 1$; $P_5 = 8,1$) в стек вводят элементы предыдущей матрицы ($1 = C_4$; $0 = C_3$; $1 = C_2$; $0 = C_1$), снова выполняют программу ($P_7 = 8,2$; $P_8 = 2$; $P_4 = 8,2$; $P_5 = 2$), вводят элементы первой из заданных матриц ($5 = C_4$; $-2 = C_3$; $-3 = C_2$; $4 = C_1$) и после третьего выполнения программы в ЗУПВ хранится искомое произведение четырех матриц.

Программа 92 пригодна также для вычисления заданной степени матрицы второго порядка. Для этого возводимую в степень матрицу заносят в стек и ЗУПВ как для умножения матриц. После выполнения программы в стеке сохранится исходная матрица, а в регистрах 7, 8, 4 и 5 ЗУПВ окажется занесенной введенная в квадрат матрица. Каждый повторный пуск программы увеличивает на единицу степень записываемой в ЗУПВ матрицы.

С помощью программы 92 можно вычислять произведение квадратных матриц с комплексными элементами

$$\begin{bmatrix} c_{11r} + jc_{11i} & c_{12r} + jc_{12i} \\ c_{21r} + jc_{21i} & c_{22r} + jc_{22i} \end{bmatrix} = \\ = \begin{bmatrix} a_{11r} + ja_{11i} & a_{12r} + ja_{12i} \\ a_{21r} + ja_{21i} & a_{22r} + ja_{22i} \end{bmatrix} \begin{bmatrix} b_{11r} + jb_{11i} & b_{12r} + jb_{12i} \\ b_{21r} + jb_{21i} & b_{22r} + jb_{22i} \end{bmatrix}.$$

В этом случае после занесения в УЗП программы 92 следует руководствоваться следующим алгоритмом:

1. Занести вещественные составляющие матриц A и B соответственно в регистры стека и ЗУПВ рассмотренным ранее образом и вычислить произведение $C_{rr} = A_r B_r$, записав результат в вычислительный бланк.

2. Ввести в регистры 7, 8, 4 и 5 мнимые составляющие соответствующих элементов матрицы B и вычислить произведение $C_{rl} = A_r B_l$, записав результат в вычислительный бланк.

3. Ввести в регистры $C_4 - C_1$ мнимые составляющие элементов матрицы A , а в регистры 7, 8, 4 и 5 ЗУПВ вещественные составляющие элементов матрицы B и вычислить произведение $C_{li} = A_l B_l$, переписав результат в вычислительный бланк.

4. Ввести в регистры 7, 8, 4 и 5 мнимые составляющие матрицы B и вычислить произведение $C_{ir} = A_i B_r$, переписав результат в бланк.

5. Вычислить составляющие матриц $C_r = C_{rr} - C_{ii}$ и $C_l = C_{rl} + C_{ir}$, по которым составить матрицу $C = C_r + jC_l$.

Полуавтоматическое вычисление произведения матриц до третьего порядка выполняется с помощью программы 93. Перед пуском этой программы для вычисления произведения матрицы A на матрицу B элементы первой из них заносят в регистры памяти: $a_{11} = C_1$, $a_{12} = C_2$, $a_{13} = C_3$, $a_{21} = C_4$, $a_{22} = C_5$, $a_{23} = C_6$, $a_{31} = P_2$, $a_{32} = P_3$, $a_{33} = P_4$. Если матрица A содержит нулевые элементы или число ее строк меньше трех, то в соответствующие регистры памяти заносятся нули.

Программа 93. Вычисление произведения матриц третьего порядка

$$\begin{array}{ccccccccc} F2 & \leftarrow & P2 & \uparrow & F5 & \times & P8 & F3 & \leftarrow \\ \times & \uparrow & F8 & + & P8 & F4 & \leftarrow & P4 & \uparrow \\ F8 & + & C/P & B/P & P0 & & & & \end{array} \begin{array}{c} F6 \\ \times \\ \uparrow \end{array}$$

Для умножения матриц в регистры 5, 6 и 7 ЗУПВ вводят элементы первого столбца матрицы B ($b_{11} = P_5$, $b_{21} = P_6$, $b_{31} = P_7$), после чего трижды выполняют программу 93 (для первого пуска нажимают клавиши В/О и С/П), регистрируя выводимые на индикатор значения элементов c_{11} , c_{21} и c_{31} произведения. Затем в регистры 5, 6 и 7 вводят элементы следующего столбца матрицы B и снова трижды выполняют программу для вычисления элементов второго столбца матрицы C . Аналогично вычисляют элементы третьего столбца матрицы-произведения C .

Для полуавтоматического вычисления произведения матриц $C = AB$ до шестого порядка включительно можно воспользоваться программой 94.

Программа 94. Вычисление произведения матриц до порядка $n \leq 6$

$$\begin{array}{ccccccccc} P7 & 1 & 0 & P4 & 1 & P3 & 0 & P8 & F7 & \uparrow & F5 & \leftarrow \\ P5 & \times & \uparrow & F8 & + & P8 & F3 & 1 & + & P3 & F2 & - \\ x \neq 0 & 5 & F4 & \uparrow & F3 & + & C/P & B/P & 1 & F5 & \leftarrow & P5 \\ F3 & 1 & + & P3 & 8 & - & x=0 & 5 & F4 & 1 & 0 & + \\ P4 & F8 & C/P & P7 & B/P & F & \uparrow & & & & & \end{array}$$

Перед пуском программы 94 в стек вводят элементы очередного j -го столбца матрицы B размера $n \times p$ ($n \leq 6$), а в регистр X — очередные элементы i -й строки матрицы A . При $j < n$ на индикаторе высвечивается индекс ij следующего элемента, а при $j = n$ — значение вычисляемого элемента c_{ij} . Подобные операции повторяются согласно следующему алгоритму:

1. Записать в регистр 2 число $n = P2$ столбцов матрицы A . Принять $j = 1$ и перейти к п. 2.

2. Ввести в стек элементы j -го столбца матрицы B в следующем порядке: $b_{1j} = C1$, $b_{2j} = C2$, ..., $b_{nj} = CN$. Принять $i = 1$, $k = 1$ и перейти к п. 3.

3. Ввести в регистр X элемент a_{ik} и нажать клавишу С/П (при вводе a_{1k} нажать клавиши В/О и С/П). После выполнения программы перейти к п. 4.

4. Если $k = n$, то записать вычисляемое число c_{ij} и перейти к п. 5, иначе принять $k = k + 1$ и перейти к п. 3.

5. Если $i = m$, то перейти к п. 6, иначе принять $i = i + 1$, $k = 1$ и перейти к п. 3.

6. Если $j = p$, то закончить вычисления, иначе принять $j = j + 1$ и перейти к п. 2.

Правильность введения в УЗП программы 94 можно проверить по контрольному примеру

$$\begin{bmatrix} 1 & 6 & -2 & 3 \\ 2 & -2 & 5 & 1 \\ -1 & 4 & 3 & 2 \\ 5 & 0 & -5 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 3 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 16 \\ 13 \\ 11 \\ 15 \end{bmatrix}.$$

Полуавтоматическое умножение двух матриц размером $m \times n$ и $n \times p$ при $6 \leq n \leq 12$ обеспечивается некоторым упрощенной программой 95.

Программа 95. Вычисление произведения векторов размером $6 \leq n \leq 12$

↑	C _x	P8	ПП	5	ПП	5	ПП	5	ПП	5	ПП
5	ПП	5	F2	ПП	6	F3	ПП	6	F4	ПП	6
F5	ПП	6	F6	ПП	6	F7	ПП	6	XY	←	↑
→	XY	←	×	↑	F8	+	P8	C/P	↑	B/O	

При использовании программы 95 в регистры памяти вводят элементы первого столбца матрицы B : $b_{11} = C1$, $b_{12} = C2$, $b_{13} = C3$, $b_{14} = C4$, $b_{15} = C5$, $b_{16} = C6$, $b_{17} = P2$, $b_{18} = P3$, $b_{19} = P4$, $b_{1,10} = P5$, $b_{1,11} = P6$, $b_{1,12} = P7$. Затем поочередно вводят элементы i -й строки матрицы A в регистр X и нажимают клавиши В/О, С/П после ввода первого элемента строки a_{ii} и только клавишу С/П после ввода последующих. Каждое выполнение программы оканчивается

выводом на индикатор значения $\sum_{k=1}^p a_{ik} b_{ki}$, которое после ввода

последнего элемента строки оказывается равным элементу c_{ii} матрицы-произведения. Окончив вычисления элементов c_{ii} первого столбца матрицы C , в регистры памяти вводят элементы второго столбца матрицы B и вычисляют элементы c_{i2} второго столбца матрицы C . Аналогично вычисляют элементы остальных ее столбцов.

В процессе решения многих задач возникает необходимость в обращении квадратных матриц. Для матрицы второго порядка с ве-

щественными элементами эта операция обеспечивается программой 96 ($a_{11} = P7$, $a_{12} = P8$, $a_{21} = P4$, $a_{22} = P5$).

Программа 96. Обращение матрицы второго порядка

F4	↑	F8	×	P2	F7	↑	F5	×	↑	F2	—
P2	1/x	↑	F7	←	F5	×	P7	→	×	P5	F4
×	/—/	P4	F8	×	/—/	P8	C/P	BП	P0		

После выполнения программы 96 (время счета около 5 с) элементы обратной матрицы хранятся в тех же регистрах, что и соответствующие элементы исходной. Поэтому правильность занесения программы и исходных данных можно проверить двойным обращением заданной матрицы $A = (A^{-1})^{-1}$ с учетом погрешностей округления. Если элементы матрицы состоят из одной-двух значащих цифр, то обращение матрицы второго порядка проще выполнять в обычном режиме вычислений, используя программу 96 лишь для матриц, элементы которых являются многозначными числами и их ввод в микрокалькулятор связан с большим числом нажатий клавиш.

Обращение матрицы третьего порядка можно выполнить автоматически с помощью программ 97 и 97а. Если столбцы заданной матрицы можно переставить так, чтобы первый не содержал нулевых элементов, то после нормирования элементов матрицы с переставленными столбцами для приведения ее к виду

$$\begin{bmatrix} 1 & b_{12} & b_{13} \\ 1 & b_{22} & b_{23} \\ 1 & b_{32} & b_{33} \end{bmatrix}$$

нормированную матрицу обращают с помощью программы 97.

Программа 97. Обращение матрицы третьего порядка

←	P2	←	P3	←	P6	←	↑	←	P7	×	P4
F7	←	P8	—	P5	F8	↑	←	F6	×	↑	F4
—	P4	F7	↑	F6	—	P6	↑	F3	×	P8	F2
↑	←	F3	←	F5	×	↑	F8	+	↑	F4	+
1/x	↑	F4	×	P4	F5	×	P5	F6	×	P6	C/P

Элементы заданной матрицы с переставленными столбцами удобно нормировать непосредственно при их вводе в микрокалькулятор с помощью предложения

$$| a_{11} | 1/x | \uparrow | a_{12} | \times | \leftarrow | a_{13} | \times | \leftarrow | a_{21} | 1/x | \uparrow | a_{22} | \times | \leftarrow | a_{23} | \times | \leftarrow | a_{31} | 1/x | \uparrow | a_{32} | \times | \leftarrow | a_{33} | x | \leftarrow |.$$

В результате элементы нормированной матрицы заносятся в стек ($b_{12} = C1$, $b_{13} = C2$, $b_{22} = C3$, $b_{23} = C4$, $b_{32} = C5$, $b_{33} = C6$). После этого трижды выполняется программа 97 (для ее пуска каждый раз нажимают клавиши В/О и С/П), вычисляющая элементы очередного столбца матрицы, обратной исходной нормированной.

Результаты каждого вычисления заносятся в регистры $P4 = \Delta_{11}/\Delta$, $P5 = \Delta_{12}/\Delta$ и $P6 = P8 = \Delta_{13}/\Delta$.

Денормирование элементов вычисленной матрицы целесообразно выполнять делением на соответствующий коэффициент a_{11} в обычном режиме вычислений. Для этого после каждого выполнения программы следует ввести предложение $|a_{11}| 1/x | \uparrow | F4 | \times | () | F5 | \times | () | F6 | \times | ()$ и записать высвечиваемые денормированные значения элементов.

Правильность использования программы 97 можно проверить по результату вычислений

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 5 & 6 \\ 1 & 9 & 11 \end{bmatrix}^{-1} = \begin{bmatrix} 0,3333333 & 1,666666 & -0,9999999 \\ -1,666666 & 2,666666 & -0,9999999 \\ 1,333333 & -2,333333 & 0,9999999 \end{bmatrix}.$$

Если обращаемая матрица разрежена и каждый ее столбец содержит по крайней мере один нулевой элемент (что исключает использование программы 97), то перестановкой столбцов ее можно привести к виду

$$\begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{21} & 0 & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix}$$

и использовать для обращения такой матрицы программу 97а.

Программа 97а. Обращение разреженной матрицы третьего порядка

$$\begin{array}{ccccccccc} F3 & \uparrow & F7 & \times & \uparrow & \rightarrow & F4 & \times & P8 \\ \times & \rightarrow & F6 & \times & \uparrow & \rightarrow & F2 & \times & \uparrow & F8 \\ F3 & \uparrow & F6 & \times & /-/- & \rightarrow & F4 & XY & \times & \rightarrow & F7 & \times \\ \rightarrow & F5 & \uparrow & F7 & \times & /-/- & \uparrow & F8 & \div & C/\Pi & F2 & \uparrow \\ F4 & \times & /-/- & P4 & F6 & \times & \rightarrow & \uparrow & F4 & P6 & B\Pi & F7 \end{array}$$

После занесения в УЗП программы 97а элементы исходной матрицы, приведенной к виду с диагональными нулевыми элементами, вводят в регистры ЗУПВ: $a_{12} = P2$, $a_{13} = P3$, $a_{21} = P4$, $a_{23} = P5$, $a_{31} = P6$, $a_{32} = P7$. Каждое выполнение программы (для первого пуска нажимают клавиши В/О и С/П) оканчивается высвечиванием очередного элемента обратной матрицы в следующем порядке: Δ_{11}/Δ , Δ_{21}/Δ , Δ_{31}/Δ , Δ_{12}/Δ , Δ_{22}/Δ , Δ_{32}/Δ , Δ_{13}/Δ , Δ_{23}/Δ , Δ_{33}/Δ . Значение определителя обращаемой матрицы хранится в регистре 8. Для проверки можно использовать результат обращения матрицы

$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 0 & 4 \\ 5 & 6 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} -0,4285714 & 0,2142857 & 0,07142857 \\ 0,3571428 & -0,1785714 & 0,10714280 \\ 0,3214285 & 0,08928571 & -0,05357142 \end{bmatrix}.$$

На микрокалькуляторе типа «Электроника Б3-34» обращение ненормированной матрицы размером 3×3 можно выполнить с помощью программы 976/34 ($a_{11} = P7, a_{12} = P8, a_{13} = P9, a_{21} = P4, a_{22} = P5, a_{23} = P6, a_{31} = P1, a_{32} = P2, a_{33} = P3$).

Программа 976/34. Обращение матрицы третьего порядка

```

ИП5 ИП3 × ИП6 ИП2 × — ЗПА ИП9 ИП2
× ИП8 ИП3 × — ЗПВ ИП8 ИП6 × ИП9
ИП5 × — ЗПС ИП1 × ИП4 ИПВ × +
ИП7 ИПА × + ЗПД ИПА ИПД ÷ С/П ИПВ
ИПД ÷ С/П ИПС ИПД ÷ С/П ИП9 ИП8 ИП7
ЗП9 → ЗП7 → ЗП8 ИП6 ИП5 ИП4 ЗП6 →
ЗП4 → ЗП5 ИП3 ИП2 ИП1 ЗП3 → ЗП1 →
ЗП2 БП 00

```

После ввода программы в УЗП и записи исходных данных нажимают клавиши В/О и С/П, что приводит к вычислению коэффициента b_{11} обратной матрицы, выводимого на индикатор. Затем, нажимая только клавишу С/П, последовательно получают значения коэффициентов b_{12}, b_{13}, b_{21} и т. д., выводимые на индикатор.

Приведенные выше программы можно использовать и при обращении матриц большого порядка по блокам в соответствии с формулами (93). При решении некоторых задач достаточно вычислить лишь определитель матрицы и миноры столбца ее элементов. Для матрицы третьего порядка решение этой задачи обеспечивается программой 98 ($a_{11} = C6, a_{21} = C5, a_{31} = C4, a_{12} = C3, a_{22} = C2, a_{32} = C1, a_{13} = P8, a_{23} = P5, a_{33} = P2$).

Программа 98. Вычисление определителя и миноров матрицы с $n = 3$

```

→ P4 → → ↑ → XY × P6 XY → ПП
P8 F5 × P3 XY P5 → ПП РВП F8 × XY
P8 F3 — P3 ПП РВП → F2 × XY P2 F3
+ С/П → P4 → ↑ → → XY × P6 XY
→ P7 ↑ F4 × ↑ F6 — ↑ F7 → В/О

```

По программе 98 вычисляют определитель матрицы, выводимый на индикатор, и три минора, заносимых в регистры $P8 = M_{13}, P5 = M_{23}, P2 = M_{33}$. При необходимости вычисления других трех миноров следует предварительно переставить на первое место соответствующую строку матрицы (с учетом возможного изменения знаков определителя и миноров, вычисляемых по программе).

Решение системы уравнений с вещественными коэффициентами

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

можно запрограммировать, используя в соответствии с формулами (80) соотношения $x_1 = (a_{22}q_1 - a_{12}q_2)/\Delta$, $x_2 = (a_{11}q_2 - a_{21}q_1)/\Delta$, где $\Delta = a_{11}a_{22} - a_{12}a_{21}$. Подобный алгоритм реализован программой 99 ($q_1 = P2$, $q_2 = P3$, $a_{11} = P7$, $a_{12} = P8$, $a_{21} = P4$, $a_{22} = P5$).

Программа 99. Решение системы из двух линейных уравнений

F8	\uparrow	F4	\times	P6	F7	\uparrow	F5	\times	\uparrow	F6	—
P6	F8	\uparrow	F3	\times	\leftarrow	F5	\uparrow	F2	\times	\uparrow	\rightarrow
—	\uparrow	F6	\div	\rightarrow	F4	\uparrow	F2	\times	\rightarrow	F7	\uparrow
F3	\times	\uparrow	\leftarrow	—	\uparrow	F6	\div	P3	\leftarrow	P2	F6
C/П БП	P0										

После выполнения программы 99 (время счета около 10 с) значение определителя матрицы коэффициентов заданной системы уравнений хранится в регистрах 6 и X, а в регистрах 2 и 3 соответственно — искомые значения x_1 и x_2 . Коэффициенты системы уравнений сохраняются в исходных регистрах, что позволяет повторить вычисления при других значениях переменных q_1 и q_2 (являющихся переменными воздействия в физических задачах).

Систему уравнений с комплексными коэффициентами

$$\begin{bmatrix} a_{11} + jb_{11} & a_{12} + jb_{12} \\ a_{21} + jb_{21} & a_{22} + jb_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

можно решить с помощью программы 100 и программ 22 или 23 для деления комплексных чисел.

Программа 100. Вычисление определителя матрицы второго порядка с комплексными элементами

C _x	P7	P8	F4	ПП	РВП	F5	ПП	F6	F2	ПП	F6
F3	ПП	РВП	F6	\leftarrow	\leftarrow	\leftarrow	F5	ПП	F8	F4	ПП
F8	F3	ПП	P8	F2	ПП	P8	F6	\leftarrow	\leftarrow	\leftarrow	F7
C/П	/—/	\uparrow	F6	\leftarrow	P6	\times	\uparrow	F7	+	P7	B/O
/—/	\uparrow	F6	\leftarrow	P6	\times	\uparrow	F8	+	P8	B/O	

После выполнения программы 100 ($a_{11} = C1$, $b_{11} = C2$, $a_{12} = C3$, $b_{12} = C4$, $a_{21} = P2$, $b_{21} = P3$, $a_{22} = P4$, $b_{22} = P5$) вещественная и мнимая части определителя хранятся соответственно в регистрах 7 и 8. Время счета около 20 с. Используя программу 24 или 25, можно в соответствии с формулами (80) вычислить искомые значения переменных x_1 и x_2 .

Систему из трех уравнений с вещественными коэффициентами можно решить по программе 101, реализующей итерационный метод Зейделя.

Программа 101. Решение системы из трех линейных уравнений с вещественными коэффициентами методом Зейделя

PП	P7	F6	PП	FBП	F7	PП	F8	F2	+	P5	PП
P7	F5	PП	FBП	F7	PП	F8	F3	+	P6	PП	P7
F5	PП	FBП	F6	PП	F8	F4	+	↑	F7	XY	P7
—	x=0	P0	C/P	×	P8	←	↑	→	XY	←	↑
B/O	×	↑	F8	+	↑	B/O					

Заданную систему предварительно необходимо преобразовать к нормальному виду

$$\begin{aligned}x_1 &= \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3; \\x_2 &= \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3; \\x_3 &= \beta_3 + \alpha_{31}x_1 + \alpha_{32}x_2.\end{aligned}$$

При заданном нулевом приближении (обычно принимают $x_i^{(0)} = \beta_i$) последующие приближения вычисляются автоматически до выполнения условия максимальной точности $x_3^{(k+1)} - x_3^{(k)} = 0$.

Исходные данные для программы 101 размещаются следующим образом: $\beta_1 = P2$, $\beta_2 = P3$, $\beta_3 = P4$, $x_1^{(0)} = P5$, $x_2^{(0)} = P6$, $x_3^{(0)} = P7$, $\alpha_{12} = C1$, $\alpha_{13} = C2$, $\alpha_{21} = C3$, $\alpha_{23} = C4$, $\alpha_{31} = C5$, $\alpha_{32} = C6$. В процессе вычислений текущие значения вычисляемых переменных заносятся на места нулевых приближений и после выполнения программы на индикаторе высвечивается 0, а искомые значения переменных хранятся в регистрах 5—7. Время счета зависит от значений коэффициентов и составляет от одной до нескольких минут.

Итерационный метод обеспечивает достаточно высокую точность результатов даже при плохо обусловленной матрице коэффициентов. Однако не всегда легко преобразовать заданную систему уравнений к виду, удовлетворяющему условию сходимости вычислений для метода Зейделя. В последнем случае для решения системы из трех линейных уравнений с вещественными коэффициентами целесообразно использовать программу 102 или 103.

Если хотя бы одна искомая переменная входит во все три уравнения заданной системы, то можно использовать программу 102, предварительно преобразовав систему к виду

$$\begin{aligned}x_1 + \alpha_{12}x_2 + \alpha_{13}x_3 &= \beta_1; \\x_1 + \alpha_{22}x_2 + \alpha_{23}x_3 &= \beta_2; \\x_1 + \alpha_{32}x_2 + \alpha_{33}x_3 &= \beta_3.\end{aligned}$$

После занесения в УЗП программы 102 вводят исходные данные: $\alpha_{12} = P2$, $\alpha_{13} = P3$, $\beta_1 = P4$, $\alpha_{23} = C1$, $\beta_2 = C2$, $\alpha_{32} = C3$, $\alpha_{33} = C4$, $\beta_3 = C5$, $\alpha_{22} = PX$.

Программа 102. Решение нормированной системы из трех линейных уравнений

ПП	РВП	ПП	РВП	\leftarrow	\leftarrow	\uparrow	F7	$-$	\leftarrow	\uparrow	F8
$\frac{1}{-}$	\uparrow	\rightarrow	$\frac{1}{-}$	\uparrow	F3	\times	\leftarrow	F7	\times	\leftarrow	F8
X Y	\rightarrow	$\frac{1}{-}$	F2	\times	XY	\rightarrow	$+ \uparrow$	F4	XY	\uparrow	
$\frac{1}{-}$	C/P	\uparrow	F2	$-$	P5	\leftarrow	\uparrow	F3	$-$	\uparrow	F5
$\frac{1}{-}$	P7	\leftarrow	F4	$-$	F5	\uparrow	\div	P8	\leftarrow	\uparrow	B/O

Результаты вычислений хранятся в регистрах PX = x_1 , C1 = $= x_2$, C2 = x_3 . Правильность занесения программы 102 можно проверить для системы уравнений

$$\begin{aligned}x_1 + 5x_2 + 8x_3 &= 12; \\x_1 + 4x_2 + 6x_3 &= 1; \\x_1 + 12x_2 - 2x_3 &= 0\end{aligned}$$

по результатам вычислений $x_1 = -35,58333$; $x_2 = 3,583333$; $x_3 = 3,708333$.

Если в заданной системе уравнений ни одна искомая переменная не входит во все три уравнения, то ее можно записать в виде

$$\begin{aligned}a_{12}x_2 + a_{13}x_3 &= q_1; \\a_{21}x_1 + a_{23}x_3 &= q_2; \\a_{31}x_1 + a_{32}x_2 &= q_3\end{aligned}$$

и использовать программу 102а ($a_{12} = C1$, $a_{13} = C2$, $a_{21} = C3$, $a_{23} = C4$, $a_{31} = C5$, $a_{32} = C6$, $q_1 = P2$, $q_2 = P3$, $q_3 = P4$). После нажатия клавиши B/O и C/P вычисляется значение x_3 , высвечиваемое на индикаторе, последующие нажатия клавиши C/P приводят к вычислению и высвечиванию значений x_2 и x_1 .

Программа 102а. Решение неполной системы из трех линейных уравнений

ПП	P9	P5	F2	\times	P2	ПП	P9	P6	F3	\times	P3
ПП	P9	P7	F4	\times	P4	F7	\uparrow	F5	\times	\uparrow	F6
+	\rightarrow	\uparrow	F2	\times	\uparrow	F3	$+ \uparrow$	F4	$-$	\uparrow	
\leftarrow	\div	C/P	\uparrow	F5	\times	\uparrow	F2	XY	$-$	C/P	\uparrow

Используя программу 102а для решения системы уравнений

$$2x_2 + 4x_3 = 1; 3x_1 + 8x_3 = -10; 4x_1 + x_2 = 5,$$

получаем $x_3 = -1,407894$; $x_2 = 3,315789$; $x_1 = 0,4210527$.

Для микрокалькулятора типа «Электроника Б3-34» легко составить программу решения системы, состоящей из трех линейных уравнений, не требующую ее предварительного преобразования. Примером такой программы является программа 103/34, реализу-

ющая метод Гаусса при следующем размещении исходных данных: $a_{11} = P7$, $a_{12} = P8$, $a_{13} = P9$, $a_{21} = P4$, $a_{22} = P5$, $a_{23} = P6$, $a_{31} = P1$, $a_{32} = P2$, $a_{33} = P3$, $q_1 = PA$, $q_2 = PB$, $q_3 = PC$. Результаты выполнения программы заносятся в регистры: $PX = PA = x_1$, $PB = x_2$, $PC = x_3$.

Программа 103/34. Решение системы из трех уравнений методом Гаусса

ИП7	ИП5	\times	ИП4	ИП8	\times	—	ЗП5	ИП7	ИП6
\times	ИП4	ИП9	\times	—	ЗП6	ИП7	ИПВ	\times	ИП4
ИПА	\times	—	ЗПВ	ИП7	ИП2	\times	ИП1	ИП8	\times
—	ЗП2	ИП7	ИП3	\times	ИП1	ИП9	\times	—	ЗП3
ИП7	ИПС	\times	ИП1	ИПА	\times	—	ИП5	\times	ИП2
ИПВ	\times	—	ИП5	ИП3	\times	ИП6	ИП2	\times	—
\div	ЗПС	ИПВ	ИПС	ИП6	\times	—	ИП5	\div	ЗПВ
ИПА	ИПС	ИП9	\times	—	ИПВ	ИП8	\times	—	ИП7
\div	ЗПА	С/П	БП	00					

Принимая один из свободных членов равным единице, а два других — равными нулю, программу 103/34 можно использовать для обращения матриц размером 3×3 . Однако каждое выполнение этой программы позволит найти лишь три элемента обратной матрицы, а изменение содержания регистров потребует восстановления в них исходных данных перед повторным выполнением программы. Поэтому целесообразно воспользоваться специализированной программой обращения (программа 976/34).

Решение систем уравнений порядка $n \leq 6$ можно существенно упростить, используя микрокалькулятор в полуавтоматическом режиме вычислений по схеме единственного деления с помощью программ 104—106. Программа 104 используется для вычислений при прямом ходе, программа 105 — при обратном, а программа 106 используется для вычислений невязок уравнений при подстановке результатов вычислений по программам 104 и 105.

При решении системы из $n \leq 6$ уравнений с помощью этих программ заготовляют вычислительный бланк, подобный приведенному в табл. 19, но без отмеченных строк и контрольного столбца. Коэффициенты и свободные члены уравнений записывают в начальную часть бланка как для вычислений в обычном режиме. После этого в УЗП вводят программу 104.

Программа 104. Вычисления при прямом ходе решения системы из $n \leq 6$ уравнений по схеме единственного деления

1/x	1	\leftarrow	\times	\leftarrow	\times	\leftarrow	\times	\leftarrow	\times	\leftarrow	\times
\leftarrow	\times	\leftarrow	F7	1	—	P7	C/П	/—/	P2	F7	P6
C/П	P3	F8	\leftarrow	P8	\uparrow	F2	\times	\uparrow	F3	+	C/П
P3	F6	1	—	P6	$x=0$	P,	F7	7	—	P5	F8
\leftarrow	P8	F5	1	+	P5	$x=0$	C _x	F3	БП	P \div	

Последующие операции описываются следующим алгоритмом:

1. Занести $n + 1 = P7$, принять $p = 1$ и перейти к п. 2.
2. Ввести в стек против часовой стрелки элементы первой строки p -й части бланка, включая свободный член (при числе $n < 6$ стековорачивается против часовой стрелки так, чтобы ведущий элемент a_{pp} оказался выведенным в индикаторный регистр), нажать клавиши В/О и С/П. Проверить высвечиваемое число $k = n + 1 = p$ строк p -й части бланка, принять $i = 2$ и перейти к п. 3.

3. Ввести в индикаторный регистр первый элемент i -й строки p -й части бланка, нажать клавишу С/П и проверить высвечиваемое число k строк $(p + 1)$ -й части бланка. Перейти к п. 4.

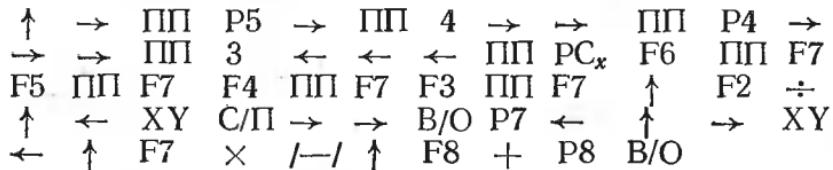
4. Ввести в индикаторный регистр следующие элементы i -й строки p -й части бланка с нажатием после каждого ввода клавиши С/П и регистрацией высвечиваемого результата $\alpha_{ij}^{(p+1)}$ в i -ю строку $(p + 1)$ -й части бланка. После регистрации всех элементов α_{ij} перейти к п. 5..

5. Если $i < k$, то принять $i = i + 1$ и перейти к п. 3, иначе к п. 6.

6. Если $k = 1$, то закончить вычисления, иначе принять $p = p + 1$ и перейти к п. 2.

Для выполнения операций обратного хода схемы единственного деления в УЗП вводят программу 105 с сохранением содержимого регистров памяти, хранящих результаты вычислений по предыдущей программе.

Программа 105. Вычисления при обратном ходе решения системы из $n \leq 6$ уравнений по схеме единственного деления



Вычисления при обратном ходе выполняются в соответствии со следующим алгоритмом:

1. Ввести в регистры 2, 3 и 8 значения элементов любой строки части бланка с тремя строками (значение $\alpha_{i,n+1}$ здесь и далее заносится в регистр 8), ввести в регистр X ранее вычисленное значение x_n , нажать клавиши В/О и С/П, записать в бланк высвечиваемое значение x_{n-1} и перейти к п. 2, приняв $k = 2$.

2. Записать в регистры 2, 3, ..., $(k + 2), 8$ элементы любой строки части бланка с k строками (кроме первой), нажать клавишу С/П и записать значение x_{n-k} . Перейти к п. 3.

3. Если $k < n$, то принять $k = k + 1$ и перейти к п. 2, иначе закончить вычисления.

После выполнения приведенного алгоритма в стеке хранятся все искомые переменные, что облегчает проверку точности решения с помощью программы 106.

Программа 106. Вычисление невязок системы уравнений при $n \leq 6$

↑	3	—	$x \neq 0$	F5	1	—	$x \neq 0$	P6	1	—	$x \neq 0$
FБП	ПП	PC _x	F6	ПП	F7	F5	ПП	F7	F4	ПП	F7
F3	ПП	F7	F2	ПП	F7	C/P	←	←	←	БП	3
←	←	БП	P3	←	БП	2	P7	←	↑	→	XY
←	↑	F7	×	/—/	↑	F8	+	P8	B/O		

Вычисление невязок при подстановке в исходную систему уравнений значений x_i , вычисленных по программам 104 и 105, описывается (после выполнения программы 105 и введения в УЗП программы 106) следующим алгоритмом:

1. Принять $i = 1$ и перейти к п. 2.
2. Ввести в регистры 2, 3, ..., ЗУПВ элементы i -й строки матрицы коэффициентов системы уравнений, а в регистр 8 — свободный член q_i . В регистр X ввести число n уравнений заданной системы, нажать клавиши B/O и C/P, записать высвеченное значение невязки σ_i и перейти к п.3.
3. Если $i = n$, то закончить вычисления, иначе принять $i = i + 1$ и перейти к п.2.

При самостоятельном использовании программы 106 для вычисления невязок в стек вводят по часовой стрелке значения переменных в порядке возрастания их индексов.

Точные методы решения системы линейных уравнений, подобные методу исключения, при плохо обусловленной матрице коэффициентов могут привести к значительным погрешностям результатов. Поэтому контроль точности решения с помощью программы 106 или ей подобной (при $n < 6$ эту программу можно существенно упростить) имеет большое значение как для контроля точности вычислений, так и для оценки особенностей моделируемого физического объекта, приводящих к плохой обусловленности матрицы коэффициентов моделирующей его свойства системы уравнений.

Для решения системы уравнений с $n \leq 6$ линейными уравнениями методом Жордана — Гаусса программа 104 пригодна без каких-либо изменений. В этом случае результаты промежуточных вычислений записывают в бланк, аналогичный используемому при вычислениях в обычном режиме (см. табл. 22), причем элементы отмеченной строки переносят в этот бланк из стека.

Программы 104—106 применимы и для решения систем с произвольным числом $n > 6$ линейных уравнений. В этом случае для вычислений с помощью программы 104 элементов каждой ($p + 1$)-й части бланка по элементам p -й части при прямом ходе выполняют следующие операции:

1. Отделяют первый столбец p -й части бланка, а остальные столбцы этой части разбивают на блоки по шесть столбцов (последний блок может иметь меньшее число столбцов).
2. Элементы каждого m -го блока ($p + 1$)-й части вычисляют согласно пп. 2—4 приведенного для программы 104 алгоритма по элементам первого столбца и m -го блока p -й части. При этом перед

вычислением элементов каждого блока в регистр 7 заносят число, на единицу большее числа строк p -й части.

Для вычисления искомых переменных при обратном ходе схемы единственного деления с помощью программы 105 при $n \geq 6$ выполняют следующие операции:

1. Переменные x_n, \dots, x_{n-5} вычисляют в соответствии с приведенным ранее для программы 105 алгоритмом по последним шести частям бланка.

2. Отделяют последние шесть столбцов бланка (контрольный столбец не учитывается) и для каждой части с номером $p < (n - 6)$ по элементам одной из строк (например, первой) вычисляют вспомогательные переменные y_{mp} таким же образом, как и переменную x_{n-5} для $(n - 6)$ -й части бланка.

3. Используя переменные y_{mp} аналогично элементам $a_{i,n+1}$, а переменную x_{n-5} как переменную x_n согласно п. 1, вычисляют переменные x_{n-6}, \dots, x_{n-12} для предыдущего блока из шести столбцов.

4. Операции пп. 2 и 3 повторяют для следующих слева блоков до вычисления всех искомых переменных x_t .

Невязки уравнений при $n > 6$ вычисляют с помощью программы 106 по блокам с последующим суммированием частных невязок для каждой строки.

Системы из произвольного числа линейных уравнений можно решать в полуавтоматическом режиме по компактной схеме исключения переменных, используя вычислительный бланк, подобный приведенному в табл. 21, и программу 107, перед пуском которой очищают регистр 5.

Программа 107. Решение системы уравнений по компактной схеме

F4	$x=0$	FXY	F2	\uparrow	F3	\div	C/P	BП	P0	F5	1
+	P5	F7	\uparrow	F8	\times	$/-$	\uparrow	F6	+	P6	F4
\uparrow	F5	$\overline{-}$	$x=0$	BП	F6	\uparrow	F2	$+$	\uparrow	F3	\div
P2	0	0	P5	P6	C/P	BП	P0				

Программой 107 реализованы вычисления по формулам (82) с помощью обобщенного выражения

$$F2 = \left(F2 - \sum_{k=1}^{F4} F7 F8 \right) / F3,$$

где каждую переменную FN в правой части перед пуском программы (первый раз — клавишами B/O и C/P) вводят в регистр ЗУПВ с соответствующим номером N . При $F4 = 0$ результат вычислений высвечивается непосредственно на индикаторе, а при $F4 > 0$ высвечивается число $F4 - k$ пар значений $F7$ и $F8$, которые должны быть введены. После ввода всех этих пар с выполнением программы после каждого ввода на индикаторе высвечивается 00 и из регистра 2 вызывают и записывают в бланк вычисленное значение переменной $F2$.

При вычислениях по формуле (82а) $a_{11} = l_{11} = P3,0 = P4$ и после ввода значений $a_{1j} = P2$ и выполнения программы высвечиваются значения u_{1j} записываются в бланк. При вычислениях по формуле (82б) $a_{ij} = P2, i = P3, j - 1 = P4$ и после каждого ввода пар значений $l_{ik} = P7$ и $u_{kj} = P8$ программу выполняют до высвечивания цифр 00. Аналогично при вычислениях по формуле (82в) вводят $a_{ij} = P2, l_{ii} = P3, i - 1 = P4, l_{ik} = P7, u_{kj} = P8$, а при вычислениях по формуле (82г) $a_{i,n+1} = P2, 1 = P3, n - i - 1 = P4, u_{ik} = P7, x_k = P8$. Время счета при $F4 = 0$ около 3 с, при $F4 > 0$ около 8 с.

Пример 55. Решить систему уравнений

$$\left[\begin{array}{cccccc} 1 & 3 & -2 & 0 & 4 & 10 \\ 1,5 & -2 & 3 & 4 & 0 & 6 \\ 1 & 1 & 0 & 1 & -2 & 4 \\ 16 & -12 & -0,2 & 3 & -1 & -1 \\ 3 & -2 & 4 & -5 & 6 & -6 \\ 1 & 0 & -1 & 0 & 1 & 0 \end{array} \right] = \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \right] = \left[\begin{array}{c} 25 \\ -3 \\ 0 \\ 1 \\ 7 \\ 3 \end{array} \right].$$

Заготовив вычислительный бланк для решения заданной системы уравнений по схеме единственного деления и заполнив его начальную часть исходными данными, выполним вычисления по программам 104 и 105 для прямого и обратного хода, и запишем результаты в бланк. Вычислив по программе 106 невязки уравнений при подстановке вычисленных значений искомых переменных, убедимся в достаточно высокой точности полученных результатов (табл. 51).

Для сравнения решим заданную систему уравнений методом разбиения на блоки. Матрица коэффициентов заданной системы недостаточно разрежена для эффективного использования метода окаймления с вычислениями по формулам (94). Поэтому ищем решение, непосредственно используя преобразования (89) с предварительным понижением порядка системы.

Перенесем третье уравнение на последнее место и исключим из остальных уравнений переменную x_3 , используя формулу (90):

$$\left[\begin{array}{cccccc} -1 & 3 & 0 & 2 & 10 \\ 4,5 & -2 & 4 & 3 & 6 \\ 15,8 & -12 & 3 & -1,2 & -1 \\ 7 & -2 & -5 & 10 & -6 \\ 1 & 1 & 1 & -2 & 4 \end{array} \right] = \left[\begin{array}{c} x_1 \\ x_2 \\ x_4 \\ x_5 \\ x_6 \end{array} \right] = \left[\begin{array}{c} 19 \\ 6 \\ 0,4 \\ 19 \\ 0 \end{array} \right].$$

Исключая переменную x_4 и используя последнее уравнение, получаем

$$\left[\begin{array}{ccccc} -1 & 3 & 2 & 10 \\ 0,5 & -6 & 11 & -10 \\ 12,8 & -15 & 4,8 & -13 \\ 12 & 3 & 0 & 14 \end{array} \right] = \left[\begin{array}{c} x_1 \\ x_2 \\ x_5 \\ x_6 \end{array} \right] = \left[\begin{array}{c} 19 \\ 6 \\ 0,4 \\ 19 \end{array} \right].$$

Разбивая эту систему уравнений на два блока по два уравнения, составляем для вектора $X_a = [x_1 x_2]^t$ согласно формуле (94) расчетное соотношение

$$X_a = (A_{aa} - A_{ab}A_{bb}^{-1}A_{ba})^{-1}(Q_a - A_{ab}A_{bb}^{-1}Q_b).$$

По программе 96 или в обычном режиме вычислений находим

$$A_{bb}^{-1} = \begin{bmatrix} 4,8 & -13 \\ 0 & 14 \end{bmatrix}^{-1} = \begin{bmatrix} 0,20833333 & 0,19345238 \\ 0 & 0,071428571 \end{bmatrix}.$$

Таблица 51

p	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	q_i	\bar{q}_i
1	1 1,5 1 16 3 1	3 -2 1 -12 -2 0	-2 3 0 -0,2 4 -1	0 4 1 3 -5 0	4 0 -2 -1 6 1	10 6 4 -1 -6 0	25 -3 0 1 7 3	0 0,00000065 -0,0000001 -0,0007674 -0,0000765 -0,0000580
2		-6,5 -2 -60 31,8 -11 -3	6 2 1 3 -5 0	4 1 -65 3 -5 -3	-6 -6 -65 -6 -3 -3	-9 -6 -161 -36 -10	40,5 -25 -399 -68 -22	
3			0,1538462 -23,58461 -0,153845 -1,769230	-0,2307692 -33,92307 -11,76923 -1,846153	-4,153845 -9,615386 4,153845 -0,2307693	-3,230769 -77,92808 -20,76923 -5,846154	-12,53846 -25,15386 0,538459 -3,307693	
4				-69,29996 -11,99999 -4,499996	-646,3996 -0,0000316 -47,99996	-573,1997 -23,99997 -42,99996	-1947,298 -11,99990 -147,4998	
5					111,9306 -6,025976	75,25535 -5,779215	325,1944 -21,05193	
6						-1,727714	-3,544532	
x_j	-0,3964561	-1,654746	-1,870545	-3,103152	1,525968	2,051573		

С помощью программы 92 (суммируя и обращая матрицы в обычном режиме) вычисляем

$$A_{ab} A_{bb}^{-1} = \begin{bmatrix} 2 & 10 \\ 11 & -10 \end{bmatrix} \begin{bmatrix} 0,20833333 & 0,19345238 \\ 0 & 0,071428571 \end{bmatrix} = \begin{bmatrix} 0,41666666 & 1,1011904 \\ 2,29166666 & 1,4136904 \end{bmatrix};$$

$$\begin{aligned} A_{ab} A_{bb}^{-1} A_{ba} &= \begin{bmatrix} 0,41666666 & 1,1011904 \\ 2,29166666 & 1,4136904 \end{bmatrix} \begin{bmatrix} 12,8 & -15 \\ 12 & 3 \end{bmatrix} = \\ &= \begin{bmatrix} 18,547617 & -2,9464287 \\ 46,297616 & -30,133928 \end{bmatrix}; \end{aligned}$$

$$A = (A_{aa} - A_{ab} A_{bb}^{-1} A_{ba})^{-1} = \left\{ \begin{bmatrix} -1 & 3 \\ 0,5 & -6 \end{bmatrix} - \begin{bmatrix} 18,547617 & -2,9464287 \\ 42,297616 & -30,133928 \end{bmatrix} \right\}^{-1} =$$

$$= \begin{bmatrix} -19,547617 & 5,9464287 \\ -45,797616 & 24,13392 \end{bmatrix}^{-1} = \begin{bmatrix} -0,12101542 & 0,029817342 \\ -0,22964425 & 0,098018157 \end{bmatrix};$$

$$\begin{aligned} B = Q_a - A_{ab} A_{bb}^{-1} Q_b &= \begin{bmatrix} 19 \\ 6 \end{bmatrix} - \begin{bmatrix} 0,41666666 & 1,1011904 \\ 2,29166666 & 1,4136904 \end{bmatrix} \begin{bmatrix} 0,4 \\ 19 \end{bmatrix} = \\ &= \begin{bmatrix} -2,0892840 \\ -21,776784 \end{bmatrix}; \end{aligned}$$

$$\begin{aligned} X_a = AB &= \begin{bmatrix} -0,12101542 & 0,029817342 \\ -0,22964425 & 0,098018157 \end{bmatrix} \begin{bmatrix} -2,0892840 \\ -21,776784 \end{bmatrix} = \\ &= \begin{bmatrix} -0,39649023 \\ -1,6547281 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \end{aligned}$$

Для решения второго блока вычисляем

$$\begin{aligned} X_b = A_{bb}^{-1} (Q_b - A_{ba} X_a) &= A_{bb}^{-1} \left\{ \begin{bmatrix} 0,4 \\ 19 \end{bmatrix} - \begin{bmatrix} 12,8 & -15 \\ 12 & 3 \end{bmatrix} \begin{bmatrix} -0,39649023 \\ -1,6547281 \end{bmatrix} \right\} = \\ &= \begin{bmatrix} 0,20833333 & 0,19345230 \\ 0 & 0,071428571 \end{bmatrix} \begin{bmatrix} -19,345846 \\ 28,722067 \end{bmatrix} = \begin{bmatrix} 1,5259677 \\ 2,0515762 \end{bmatrix} = \begin{bmatrix} x_5 \\ x_6 \end{bmatrix}. \end{aligned}$$

По исключенным уравнениям вычисляем

$$x_3 = x_1 + x_5 - 3 = -1,8705225; x_4 = 2x_5 - 4x_6 - x_1 - x_2 = -3,1031511.$$

Подстановкой вычисленных значений искомых переменных в исходную систему уравнений вычисляем невязки $\delta_1 = 3 \cdot 10^{-6}$; $\delta_2 = 6 \cdot 10^{-6}$; $\delta_3 = 0$; $\delta_4 = 1,1 \times 10^{-6}$; $\delta_5 = 0$; $\delta_6 = 0$. Таким образом, решение по схеме единственного деления характеризуется большей погрешностью результатов, чем решение разбиением на блоки.

При использовании программы 107 вычислительный бланк имеет меньшие размеры, чем при схеме единственного деления, но затраты времени при использовании программ 104—106 обычно меньше за счет большей степени автоматизации вычислений.

Программы 104—107 можно использовать для вычисления определителей матриц произвольного порядка или обращения последних. В первом случае достаточно ограничиться вычислениями прямого хода, а при обращении матриц используют вычислительные бланки, аналогичные приведенным в гл. 2 для использования микрокалькулятора в обычном режиме.

2. РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Нелинейные уравнения обычно решают приближенными численными методами, так как аналитически определить корни таких уравнений за редкими исключениями невозможно. Процедура решения при численных методах состоит из двух этапов:

1. Определение интервала (или области на комплексной плоскости), в котором находится корень уравнения (этап отделения корней).

2. Итерационное сокращение этого интервала (этап уточнения корня) до тех пор, пока не будет достигнута требуемая точность для ширины интервала $\delta x = |x_k - x_{k-1}| \leq \varepsilon_1$ или невязки уравнения $\delta = |f(x_k)| \leq \varepsilon_2$.

Решению уравнения, как правило, предшествует его анализ для оценки пригодности или эффективности выбранного метода решения и выбора оптимального или близкого к оптимальному алгоритма итерационной процедуры. Иногда целесообразно преобразовать исходное уравнение для упрощения вычислений или сокращение времени счета. В процессе анализа условий задачи часто удается получить грубую оценку интервала, заключающего искомый корень. Во многих случаях полезно приближенное графическое представление анализируемого уравнения.

Интервал нахождения корня сокращают по одному из известных алгоритмов, из которых наиболее пригодные для микрокалькуляторов рассмотрены в гл. 1. Эффективность таких алгоритмов, определяемая, главным образом, затратами усилий и времени для достижения результата с требуемой точностью, зависит от вида заданного уравнения, его поведения на интервале нахождения корня и характеристик используемого вычислительного устройства. Если при использовании микрокалькулятора в обычном режиме или ручном счете пользователь в процессе решения имеет наглядное представление о процессе сходимости и может в любой момент изменить алгоритм решения, то при автоматическом решении итерационный процесс происходит скрыто и пользователь видит лишь окончательный результат (если он достигается). Поэтому в сомнительных случаях, когда нет уверенности в сходимости процесса, целесообразно программировать решение таким образом, чтобы после каждой итерации вычисления прекращались и значение промежуточного результата высвечивалось на индикаторе. В этом случае сравнение значений x_k и x_{k-1} позволяет убедиться в пригодности выбранного алгоритма и скорости сходимости вычислительного процесса.

Рассмотрим на нескольких примерах основные алгоритмы решения, обращая внимание на такие их характеристики, как скорость сходимости, определяющую число итераций, необходимых для достижения требуемой точности, объем вычислений на каждой итерации, трудоемкость предварительного решения уравнения и сложность программирования выбранной процедуры.

Проще всего программируется метод простых итераций, при использовании которого уравнение предварительно приводится к виду уравнения (59). Сходимость обеспечивается, если на интервале между нулевым приближением x_0 и искомым корнем x_k выполняется условие $|f'(x)| < 1$, причем скорость сходимости тем больше, чем сильнее приведенное неравенство.

Точность определения корня зависит от погрешности вычисления правой части уравнения (59) и оценивается по приведенным в гл. 1 правилам. Программная реализация метода простых итераций при использовании критерия максимальной точности содержит 11 вспомогательных операторов, обеспечивающих после каждой итерации присвоение переменной x_i нового значения x_{i+1} и вычисление приращения этой переменной $\delta x_i = x_{i+1} - x_i$. В незаполненную часть программы 108, реализующей метод простых итераций, вписывается арифметическое выражение для вычисления правой части решаемого уравнения.

Программа 108. Реализация метода простых итераций

P8 F8 ... ↑F8 XY P8 — $x = 0$ F0 F8 С/П

Перед пуском программы 108, дополненной фрагментом вычисления правой части уравнения, нулевое приближение корня вводят в регистр X и автоматически заносится в регистр δ в процессе выполнения программы. В этот же регистр заносятся текущие приближения корня и его окончательное значение после выполнения программы. Остальные регистры можно использовать для хранения исходных данных, необходимых для вычисления правой части уравнения, и промежуточных результатов.

Пример 56. Зависимость тока эмиттера I_s от напряжения источника питания базы E_b биполярного транзистора (рис. 38) описывается уравнением

$$I_s \approx I_0 [e^{\Delta(E_b - (R_s + R_b(1-\alpha))I_s)} - 1].$$

Определить значение I_s при $I_0 = 0,01 \text{ mA}$; $\Delta = 39 \text{ 1/B}$; $R_s = 1 \text{ k}\Omega$; $R_b = 50 \text{ k}\Omega$; $E_b = 10 \text{ V}$; $\alpha = 0,95$.

Производная правой части заданного уравнения в части интервала аргумента $[0, 10]$, определяемого условиями задачи, по модулю превышает единицу и, следовательно, условия сходимости метода простых итераций не выполняются. Однако, прологарифмировав обе части уравнений, можно привести его к виду

$$I_s \approx \frac{E_b - (\ln(I_s/I_0 + 1))/\Delta}{R_s + (1 - \alpha)R_b},$$

при котором условие сходимости выполняется в интервале

$$I_s > \frac{1}{\Delta(R_s + (1 - \alpha)R_b)} - I_0,$$

представляющем практический интерес.

При составлении программы решения вычисление коэффициента $R_s + (1 - \alpha)R_b$ целесообразно вынести за пределы итерационного цикла. Размещая исходные данные $I_0 = P2$, $\Delta = P3$, $E_b = P4$, $R_s = P5$, $R_b = P6$, $\alpha = P7$, нетрудно на основе программы 108 составить программу 109 для решения заданного уравнения методом простых итераций.

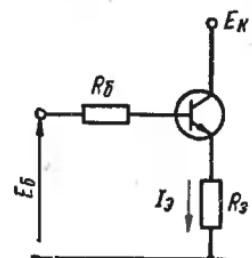


Рис. 38. Схема транзисторной ступени с нагрузкой в цепи эмиттера

Программа 109. Вычисление тока эмиттера биполярного транзистора

P8	F7	1	—	↑	F6	×	↑	F5	—	P6	F8
↑	F2	÷	1	+	ln	↑	F3	÷	↑	F4	—
↑	F6	÷	↑	F8	XY	P8	—	x=0	XY	F8	C/P

Вводя начальное приближение в индикаторный регистр, после выполнения программы (время счета около 25 с), получим на индикаторе искомое значение $I_s = 2,815795 \text{ mA}$.

Погрешность результата определяется погрешностью вычисления функции $\ln x$ (до двух единиц младшего разряда), погрешностью деления на коэффициент A (до единицы младшего разряда) и погрешностью деления всего числителя на знаменатель. С учетом функций чувствительностей к каждому источнику погрешностей получаем

$$\delta I_s \leq |10^{-7}I_s| + \left| \frac{10^{-7}}{R_9 + (1-\alpha)R_6} \right| + \left| \frac{2 \cdot 10^{-7}}{\Delta(R_9 + (1-\alpha)R_6)} \right| \approx \\ \approx 2,8 \cdot 10^{-7} + \frac{10^{-7}}{3,5} + \frac{2 \cdot 10^{-7}}{39 \cdot 3,5} \approx 3,1 \cdot 10^{-7}.$$

Метод простых итераций пригоден и для вычисления комплексных корней. Критерием сходимости в этом случае является условие $|f'(z)| < 1$ во всей области нахождения корня на плоскости комплексной переменной $z = x + jy$. Решаемое уравнение разбивается на два вещественных: для вещественной и мнимой частей исходного уравнения. Нулевое приближение представляет собой комплексное число $z_0 = x_0 + jy_0$.

Сложность анализа сходимости (или приведения уравнения к виду, при котором обеспечивается сходимость) является серьезным препятствием практического применения метода простых итераций для вычисления комплексных корней нелинейных уравнений общего вида.

Методом половинного деления можно на каждой итерации вдвое сокращать интервал, заключающий искомый корень. Программная реализация этого метода допускает несколько вариантов, один из которых представлен в виде программы 110.

Программа 110. Реализация метода половинного деления

F6	2	÷	P6	↑	F7	+	P7	÷	sin	x=0	PX
F7	C/P	...	↑	F8	XY	P8	×	x<0	P0	F6	/—/
P6	БП	P0									

В незаполненную часть этой программы записывают выражение, вычисляющее левую часть уравнения $y(x) = 0$ при значении $x = x_i$, занесим программой в регистр 7. После ввода программы в УЗП значение левой границы интервала нахождения корня и его ширину записывают соответственно в регистры 7 и 6, а в регистр 8 — любое число, совпадающее по знаку с $y(x)$ при подстановке левой границы интервала. Регистры 2—5 ЗУПВ и стек можно использовать для вычисления $y(x_i)$.

В качестве условий окончания итерационного процесса в программе 110 использован критерий максимальной точности. Требуе-

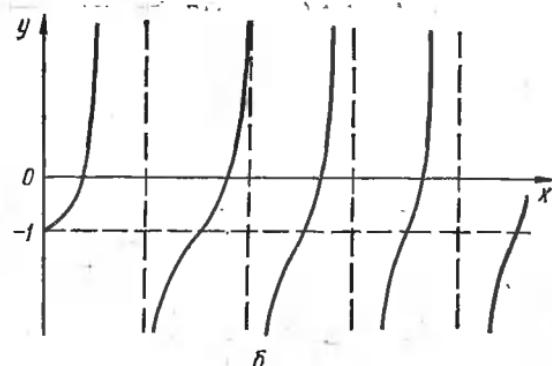
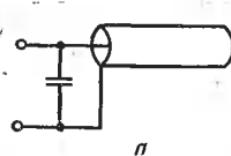
мое число итераций зависит от точности отделения корня и оценивается соотношением

$$n \leq \frac{\lg \frac{\Delta x_0}{x_i} + 8}{\lg 2} + 2,$$

где Δx_0 — начальная ширина интервала нахождения корня; x_i — искомый корень.

На практике число итераций обычно лежит в пределах $27 \leq n \leq 37$ ($x \geq 0,001 x_0$), что позволяет достаточно точно определить полное время счета по времени выполнения одной итерации. Вели-

Рис. 39. Коаксиальная линия с емкостным входом:
а — эквивалентная схема; б — график входного сопротивления



чины невязки уравнения $y(x_i)$ после окончания программы хранятся в регистре 8.

Несмотря на сравнительно медленную сходимость, метод половинного деления отличается рядом существенных преимуществ, в частности, независимостью сходимости от вида решаемого уравнения и гарантированным сходжением к корню, обеспечиваемым независимостью вычисляемого приращения Δx_i от значения $y(x_i)$. Точность результата определяется только точностью вычисления $y(x_i)$ и, следовательно, зависит от вида решаемого уравнения.

Пример 57. Вычислить частоты параллельного резонанса колебательной системы, состоящей из короткозамкнутого отрезка коаксиальной линии длиной l , заполненной диэлектриком с относительной диэлектрической проницаемостью ϵ_r , и подключенного параллельно его входу конденсатора емкостью C (рис. 39, а).

Резонансные частоты в рассматриваемой задаче при пренебрежении потерями в отрезке линии определяются уравнением

$$j\omega_p C + \frac{1}{jz_0 \operatorname{tg} \beta l} = 0,$$

где z_0 — волновое сопротивление линии; β — фазовая постоянная, вычисляемая по формуле $\beta = \omega_p / v_{\text{лин}} = \omega_p \sqrt{\epsilon_r / c}$; c — скорость света.

Переходя к нормированным единицам (C в пФ, f в МГц, z_0 в Ом, l в м) и вводя нормированную частоту $x = \pi l f \sqrt{\epsilon_r} / 150$, получаем

$$Ax \operatorname{tg} x - 1 = 0,$$

где $A = 3z_0 C / (10^4 l \sqrt{\epsilon_r})$.

Из графического представления уравнения (рис. 39, б) очевидно, что каждый корень x_k заключается в интервале шириной $\pi/2$, левая граница которого для k -го корня равна πk . Это позволяет предусмотреть в программе автоматический переход к вычислению следующей резонансной частоты после вычисления предыдущей, если выполнение программы начать с записи в регистр 7 нового значения левой границы интервала нахождения корня. С учетом этого замечания программа 110 для решения заданной задачи принимает вид программы 111.

При использовании программы 111 по исходным данным предварительно вычисляют величину A , заносимую в регистр 2, и $B = 150/(\pi \sqrt{\epsilon_r})$, записываемую в регистр 3. Первое выполнение программы по команде В/О, С/П приводит к вычислению основной резонансной частоты f_{p1} , для вычисления каждой последующей нажимают только клавишу С/П. Время счета для каждой частоты составляет 100—200 с.

Программа 111. Вычисление резонансных частот нагруженного емкостью отрезка коаксиальной линии.

C_x	P4	P7	π	2	\div	P6	1	/ — /	P8	F6	2
\div	P6	↑	F7	+	P7	\div	sin	$x=0$	P / — /	F7	↑
F3	\times	C/П	F4	↑	π	+	B/O	F7	e^{tx}	\div	↑
F7	\times	↑	F2	\times	1	—	↑	F8	XY	P8	×
$x < 0$	FXY	F6	/ — /	P6	БП	FXY					

Например, при $z_0 = 75 \text{ Ом}$; $\epsilon_r = 2,5$; $l = 1 \text{ м}$, $C = 10 \text{ пФ}$ получаем $A = 0,225/\sqrt{2,5} = 0,14230249$; $B = 150/\pi \sqrt{2,5} = 30,197527$ и по программе 111 вычисляем (в МГц) $f_{p1} = 41,58989$ ($\delta y = 3,3 \cdot 10^{-9}$); $f_{p2} = 126,1105$ ($\delta y = 1,44 \cdot 10^{-8}$) и т. д.

Если известен знак невязки $y(x)$ на правой границе интервала, содержащего искомый корень, причем этот знак не зависит от входящих в уравнение постоянных, основную программу 110 можно представить в виде программы 112.

Программа 112. Реализация метода половинного деления при известном знаке невязки на правой границе интервала

F7	2	\div	P7	↑	F8	+	P7	\div	sin	$x=0$	P X
F8	C/П	...	$x \geqslant 0$	P0	F8	↑	F7	—	P8	БП	P0

Перед использованием этой программы в регистр 7 заносят ширину интервала нахождения корня, а в регистр 8 — его левую границу. Условный оператор по адресу 84 (после ввода в программу фрагмента, вычисляющего функцию, этот адрес может измениться) определяется знаком невязки $y(x)$ на левой границе интервала: если $y(x_0 + \Delta x_0) \geqslant 0$, то записывают оператор $x \geqslant 0$, в противном случае — оператор $x < 0$.

Стратегия сокращения интервала по программе 112 отличается от используемой в программе 110: вместо смены знака переменной Δx после прохождения корня в программе 112 предусмотрен возврат в исходную точку с последующим приращением переменной на величину $\Delta x_i/2$, т. е. $x_{i+1} = x_i - \Delta x_i + \Delta x_i/2$. Программа 112 на три оператора короче программы 110, и длина фрагмента для вычисления функций может быть соответственно увеличена.

Пример 58. Напряжения U_6 и U_k в схеме нагруженного биполярного транзистора (рис. 40) в статическом режиме связаны уравнением

$$U_k + A(e^{\Delta U_6} - 1) - B(e^{\Delta U_6 - \Delta U_k} - 1) - E_k = 0,$$

где $A = 0,0095$ В; $B = 0,05$ В; $\Delta = 39$ 1/В; $E_k = 10$ В.

Определить значение U_k в интервале напряжений U_6 от 0 до 0,25 В.

При составлении программы вычисление части функции, не связанный с переменной U_k , целесообразно вынести в начало программы. Из физических соображений ясно, что возможное значение искомого напряжения ограничено интервалом $0 < U_k < 10$ В, причем при $U_k = 10$ левая часть преобразованного уравнения

$$(E_k + A - B) - Ae^{\Delta U_6} + Be^{\Delta U_6 e^{-\Delta U_k}} - U_k = 0$$

будет отрицательной.

С учетом приведенных замечаний и программы 112 составляем программу 113 ($A = 0,0095 = P2$; $B = 0,05 = P3$; $E_k = 10 = P4$; $U_6 = PX$; $PX = U_k$).

Программа 113. Определение зависимости напряжения на коллекторе от напряжения на базе биполярного транзистора

↑	3	9	×	e^x	0	P8	F3	×	P6	F2	XY
×	—	↑	F4	P7	+	↑	F3	—	P5	F7	2
÷	P7	↑	F8	+	P7	÷	sin	x=0	P6	F8	C/П
F8	3	9	×	/ — /	e^x	↑	F6	×	↑	F5	+
↑	F8	—	x<0	F÷	F8	↑	F7	—	P8	БП	F÷

Для значений $U_6 = 0; 0,05; 0,1; 0,15; 0,2; 0,25$ по программе 113 вычисляем $U_k = 9,95; 9,892728; 9,490177; 6,660774; 0,05686606; 0,04419243$. Время выполнения программы составляет около 170 с для больших напряжений на коллекторе и около 255 с для меньших напряжений.

Сходимость итерационного процесса обычно ускоряется при использовании метода хорд, описываемого формулой (56). Однако при изменении в широких пределах второй производной в интервале, содержащем корень, этот метод по быстродействию может уступать методу половинного деления. Кроме того, метод хорд чувствителен к погрешностям округления и при использовании критерия максимальной точности для прекращения вычислений возможно зацикливание программы. Программная реализация метода хорд в общем случае связана с большим числом вспомогательных операторов, что также является недостатком. Поэтому метод хорд целесообразно использовать лишь в случае, когда форма заданного уравнения позволяет упростить вычисления по формуле (56), а выбранный критерий гарантирует окончание процесса вычислений. Примером является программа 45, удовлетворяющая этим требованиям.

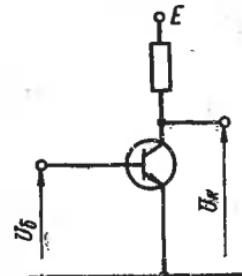


Рис. 40. Схема транзисторной ступени с нагрузкой в цепи коллектора

Метод касательных, описываемый формулой (57), обеспечивает, как правило, быструю сходимость, если на интервале нахождения корня вторая производная функции изменяется достаточно мало. В противном случае сходимость вычислений ухудшается, что увеличивает время счета. Метод касательных еще более чувствителен к ошибкам округления, чем метод хорд. Поэтому часто в качестве критерия окончания вычислений приходится выбирать более грубые условия малости приращения Δx_i . Если решаемое уравнение имеет кратные корни, то метод касательных непригоден, так как в окрестности корня, определяемой погрешностью вычислений, первая производная обращается в нуль, что приводит к переполнению при вычислении приращений функции. При сравнении метода касательных с другими методами следует учитывать, что его применение требует вычислений на каждой итерации не только значения функции, но и ее производной, хотя иногда выражение $f(x)/f'(x)$ удается упростить, сократив в результате как длину программы, так и время счета.

Основная часть программы 114, реализующей метод касательных, содержит 20 вспомогательных операторов, а остальная часть предназначена для записи операторов вычисления функции и ее первой производной.

Программа 114. Реализация метода касательных Ньютона

P8	P6	\uparrow	F6	\div	\uparrow	F8	XY
-	P8	-	x^2	V	\uparrow	F7	-	$x < 0$	F0	F8	C/P

Вычисления производной заданной функции должны быть организованы так, чтобы результат $f'(x)$ заносился в регистр 6, а текущие значения переменной x_i хранились в регистре 8. Перед пуском программы выбранное значение ε вводят в регистр 7, а начальное значение x_0 — в регистр X. По окончании вычислений значение искаемого корня хранится в регистре 8 и высвечивается на индикаторе.

Пример 59. Угол отсечки θ выходного напряжения диодного детектора связи с его параметрами ($R_{\text{пп}} = 10 \Omega$; $R_{\text{обр}} = 200 \Omega$; $R_H = 3 \text{ к}\Omega$) уравнением

$$\operatorname{tg} \theta - \theta = \pi \frac{R_{\text{пп}}}{R_H} \frac{R_{\text{обр}}}{R_{\text{пп}} + R_{\text{обр}}}.$$

Определить значение θ методом касательных.

Организовав вычисление не зависящей от θ части функции в начале программы 114, составим на ее основе программу 115 для решения заданного уравнения с вычислением производной функции по формуле $f'(\theta) = -1 + \frac{1}{1 + \cos^2 \theta}$.

Программа 115. Вычисление угла отсечки диодного детектора

P8	F3	\uparrow	F2	$+$	\div	\uparrow	F2	\times	\uparrow	F4	\div
\uparrow	π	\times	P5	F8	\cos	x^2	$1/x$	1	$-$	P6	F8
e^{ix}	\div	\uparrow	F8	$-$	\uparrow	F5	$-$	\uparrow	F6	\div	\uparrow
F8	XY	$-$	P8	$-$	x^2	V	\uparrow	F7	$-$	$x < 0$	F×
F8	C/P	\uparrow	π	\div	1	8	0	\times	C/P		

После выполнения программы 115 на индикаторе высвечивается значение угла отсечки в радианах. Для перевода этого результата в градусы достаточно дополнительно нажать клавишу С/П. Исходные данные размещаются следующим образом: $R_{\text{пр}} = P2$, $R_{\text{обр}} = P3$, $R_h = P4$, $\varepsilon = P7$, $\theta_0 = P8$.

Для заданных исходных данных и $\theta_0 = 1$ при $\varepsilon = 0,01$ получаем $\theta^{\circ} = 17,56336$ ($\theta_{\text{рад}} = 0,3065385$; $t = 50$ с); при $\varepsilon = 0,0001$ корень $\theta^{\circ} = 17,56151$ ($\theta_{\text{рад}} = 0,3065062$; $t = 60$ с); при $\varepsilon = 10^{-6}$ корень $\theta^{\circ} = 17,56151$ ($\theta_{\text{рад}} = 0,3065062$; $t = 65$ с); при $\varepsilon = 10^{-7}$ вычисления зацикливаются и результат отсутствует.

По программам, подобным программам численного решения уравнений, можно организовать поиск экстремальных точек функции одной переменной. Однако оптимальная стратегия поиска экстремума отличается от стратегии поиска корня, так как прохождение последнего приводит к изменению знака функции, тогда как при прохождении экстремума может сохраняться даже знак приращения функции (рис. 41). Это обстоятельство при использовании, например, метода половинного деления не позволяет добиться двукратного сокращения интервала нахождения экстремума на каждой итерации, так как программа поиска должна предусматривать возможность возвращения на два шага в обратном направлении после изменения приращения функции по знаку. Поэтому поиск экстремума требует больших затрат времени, чем поиск корня.

Выполнение общих операций при поиске минимума функции одной переменной методом половинного деления обеспечивается программой 116 ($x_0 = P2$, $\Delta x_0 = P3$, $P2 = x_i$).

Программа 116. Реализация одномерного поиска минимума методом половинного деления

F2	↑	F3	+	P2
...	↑	F8	XY	P8	XY	—	$x \geq 0$	P0	F3	/—/	4
÷	P3	↑	F2	XY	+	XY	—	$x = 0$	P↑	F2	C/P

Регистр 8 используется в этой программе для хранения текущих значений функции $y(x)$, вычисляемых при помощи операторов, записываемых в незаполненной части программы. При поиске по этой программе после изменения знака приращения функции шаг уменьшается в четыре раза и изменяется направление поиска для двух шагов. Это обеспечивает сокращение числа итераций вблизи точки экстремума. По программе 116 можно находить максимумы обратной функции.

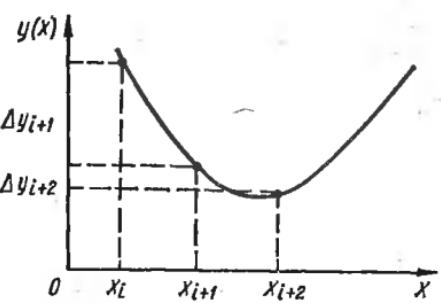


Рис. 41. График одномерного поиска минимума функции

Пример 60. Определить частоту «выброса» амплитудно-частотной характеристики фильтра нижних частот с частотной характеристикой

$$K(j\omega) = 1 \sqrt{\left[\left(1 + \frac{2\rho}{R_H Q} + j \frac{\omega}{\omega_p} \left(\frac{3}{Q} + \frac{2\rho}{R_H} + \frac{\rho}{R_H Q^2} \right) - \frac{\omega^2}{\omega_p^2} \left(3 + \frac{2\rho}{R_H Q} + \frac{1}{Q^2} \right) - j \frac{\omega^3}{\omega_p^3} \left(\frac{\rho}{R_H} + \frac{2}{Q} \right) + \frac{\omega^4}{\omega_p^4} \right] },$$

где $\rho = \sqrt{L/C}$; $Q = \rho/r$; $\omega_p = 1/\sqrt{LC}$ по следующим данным: $\omega_p = 10^7$, $Q = 50$, $\rho = 1 \text{ кОм}$, $R_H = 5 \text{ кОм}$.

Подставляя исходные данные, нормируя искомую частоту относительно резонансной как $x = \omega/\omega_p$ и заменяя функцию обратной, получаем

$$1/K(j\omega) = (1,008 - 3,0084x^2 + x^4) + jx(0,46008 - 0,24x^2)$$

или, в буквенных символах,

$$1/K(j\omega) = a + bx^2 + x^4 + jx(c + dx^2).$$

Для квадрата модуля, минимум которого совпадает с максимумом исходной функции,

$$1/|K(j\omega)|^2 = (a + bx^2 + x^4)^2 + x^2(c + dx^2)^2,$$

где $a = 1,008$; $b = -3,0084$; $c = 0,46008$; $d = -0,24$.

На основе программы 116 для полученного выражения составляем программу 117 ($a = P4$, $b = P5$, $c = P6$, $d = P7$, $y(x_0) = P8$).

Программа 117. Вычисление частот выбросов амплитудно-частотной характеристики фильтра нижних частот

F2	↑	F3	+	P2	↑	F5	+	↑	F2	×	↑
F4	+	x^2	←	F2	↑	F7	×	↑	F6	+	x^2
↑	F2	×	↑	→	+	↑	F8	XY	P8	XY	—
$x \geq 0$	P0	F3	/	—	4	÷	P3	↑	F2	XY	+
—	$x=0$	P↑	F2	γ	C/P	—	—	—	—	XY	—

Вводя в регистр 8 начальное значение $y(x_0) = 1,008^2$ при $P2 = x_0^2 = 0$; $P3 = \Delta x_0^2 = 0,5$; $0,1$; $0,02$, получаем $x_{\text{экст}} = 0,6141735$; $0,6141911$; $0,6141779$ при времени счета около 260 с.

Программа 117 оперирует с переменной $z = x^2$ и только после окончания итерационного процесса вычисляется $x = \sqrt{z}$.

Рассмотренные методы решения уравнений непригодны в тех случаях, когда корень уравнения $y(x) = 0$ совпадает с экстремальным значением функции и для его определения необходимо использовать методы математического программирования для поиска экстремумов функций. Ограниченная емкость запоминающих устройств и малое быстродействие микрокалькуляторов затрудняет решение задач математического программирования в автоматическом режиме. В тех случаях, когда такое решение возможно, для минимизации функций нескольких переменных целесообразно, как правило, использовать методы координатного спуска. Однако автоматический поиск экстремума функции даже двух переменных связан обычно со значительными затратами времени.

Для примера рассмотрим функцию двух переменных

$$\Phi(x_1, x_2) = (x_1^2 - x_2)^2 + (x_1^2 - 1)^2,$$

используемую [28] в качестве тестовой функции при проверке сходимости различных методов нелинейного программирования. Взяв за основу метод координатного спуска с комбинированным алгоритмом одновременного поиска минимума функции, составим программу 118 для определения корней рассматриваемого уравнения.

Программа 118. Минимизация тестовой функции двух переменных

ПП	F /—/	$x \geq 0$	P0	ПП	F4	$x \geq 0$	F↑	F4	4	÷	P4
ПП	F /—/	$x \geq 0$	F2	ПП	F4	$x \geq 0$	F×	F4	/—/	P4	БП
P0	F3	↑	F4	+	P3	—	$x=0$	6	C/П	F2	↑
F4	+	P2	F2	x^2	↑	←	F3	—	x^2	→	1
—	x^2	↑	←	+	↑	F8	XY	P8	XY	—	B/O

Перед пуском этой программы в регистры 2 и 3 вводят соответственно начальные значения переменных x_{10} и x_{20} , в регистр 4 — начальный шаг поиска, а в регистр 8 — начальное значение $y(x_{10}, x_{20})$.

При $x_{10} = x_{20} = 0$ и $\Delta x_0 = 0,5$ эта программа находит минимум примерно за 130 итераций, на что затрачивается около 15 мин, с результатами $x_1^* = 1; x_2^* = 0,9999999$ (точное значение корней $x_1 = x_2 = 1$); $y(x_1^*, x_2^*) = 10^{-14}$ (результаты записываются программой соответственно в регистры 2, 3 и 8 на места исходных данных).

Более удобно для решения многих задач нелинейного программирования использовать программы одномерного поиска экстремума, что позволяет оценивать результаты выполнения каждой итерации и при необходимости корректировать ход дальнейших вычислений при поиске экстремума функций двух или нескольких переменных. В этом случае наиболее просто реализуется метод покординатного спуска с поиском экстремума по одной из переменных при фиксированных значениях (начальных или полученных на предыдущих итерациях) остальных изменяемых переменных.

3. РЕШЕНИЕ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

В инженерной практике часто встречаются задачи, связанные с вычислениями корней нелинейных уравнений, левые части которых представляют собой степенные многочлены одной переменной. Составить универсальную программу решения таких алгебраических уравнений, с помощью которой можно находить как вещественные, так и комплексные корни для микрокалькулятора с ограниченной емкостью памяти, практически невозможно. Поэтому приходится использовать отдельные программы или комплекты программ для решения алгебраических уравнений различных порядков.

Решение квадратного уравнения вида $a_2x^2 + a_1x + a_0 = 0$ обеспечивается программой 119 ($a_2 = P2, a_1 = P3, a_0 = P4$).

Программа 119. Решение квадратного уравнения

F3	2	÷	↑	F2	÷	/—/	P5	F4	↑	F2	÷
XY	F5	x^2	—	P6	$x < 0$	5	/—/	$\sqrt{-}$	P6	↑	F5
XY	+	P7	—	P8	7	8	C/Π	$\sqrt{-}$	P6	x^2	
↑	F5	x^2	+	V	P7	F6	↑	F5	÷	P8	5

Если корни уравнения вещественны, то после выполнения программы 119 высвечиваются цифры 78, указывающие номера регистров 7 и 8, в которых хранятся значения корней. При высвечивании на индикаторе цифр 5678 вещественная и мнимая части комплексных корней хранятся соответственно в регистрах 5 и 6, в регистре 7 — значение модуля, а в регистре 8 — значение тангенса аргумента. Время счета по этой программе около 10 с.

Например, после решения по программе 119 уравнения $2x^2 + 4x + 10 = 0$ содержимое регистров памяти RX = 5678; P5 = = -1; P6 = 2; P7 = 2,2336067; P8 = -2 (т. е. $x_{1,2} = -1 \pm j2$; $|x_{1,2}| = 2,236067$). После решения уравнения $x^2 + 10x + 2 = 0$ по этой программе RX = 78; P7 = -0,2041685; P8 = -9,795831.

Если коэффициенты нормированного квадратного уравнения комплексны

$$x^2 + (a_1 + jb_1)x + (a_0 + jb_0) = 0,$$

то его решение можно найти по программе 120 ($a_1 = P2$, $b_1 = P3$, $a_0 = P4$, $b_0 = P5$).

Программа 120. Решение квадратного уравнения с комплексными коэффициентами

F2	x^2	↑	F3	x^2	—	4	÷	↑	F4	—	P6
F3	2	÷	P3	↑	F2	×	↑	F5	—	P7	x^2
↑	F6	x^2	+	V	↑	F6	+	2	×	$\sqrt{-}$	P8
1/x	↑.	F7	×	↑	F3	—	→	F8	↑	F2	—

2 ÷ → F8 /—/ P8 x ≥ 0 P6 0 0 C/Π

После выполнения этой программы на индикаторе высвечиваются цифры 00, а результаты вычислений хранятся в стеке: C1 = = Rex₁, C2 = Imx₁, C3 = Rex₂, C4 = Imx₂. Например, по программе 120 уравнение $x_2 + (3 + j5)x + (-4 + j8) = 0$ решается за 12 с с результатами C1 = -2; C2 = -2; C3 = -1; C4 = -3, т. е. $x_1 = -2 - j2$; $x_2 = -1 - j3$.

Кубическое уравнение $x^3 + a_2x^2 + a_1x + a_0 = 0$ решается по формулам Кардана [14], но отсутствие во входном языке операторов автоматического вычисления обратных тригонометрических функций затрудняет использование этих формул для программирования. Кроме того, погрешности извлечения корней и вычисления тригонометрических функций при использовании точных методов

решения кубических уравнений значительно увеличивают погрешности результатов.

Поэтому для решения кубических уравнений целесообразно использовать приближенный метод поиска корней с последовательным введением двух программ (121 и 122). Программа 121 обеспечивает вычисление вещественного корня кубического уравнения, противоположного по знаку свободному члену a_0 (см. приложение).

Программа 121. Вычисление вещественного корня кубического уравнения

F4 P6 F6 2 \times P6 /—/ ПП P6 $x < 0$ Р↑ F6
2 \div P6 ↑ F5 XY + P5 — $x = 0$ F4 F5
C/П ПП F6 $x \geq 0$ XY F5 ↑ F6 — P5 БП XY
P5 F5 ↑ F2 ПП Р— F3 ПП Р— F4 XY +
 \times B/O + ↑ F5 \times ↑ B/O

Процедура отделения корней в этой программе автоматизирована, для чего вначале проверяется наличие корня в интервале $[-a_0, 0]$. При отсутствии корня ширина интервала удваивается до появления в нем корня. После этого выполняется итерационный процесс сокращения интервала методом половинного деления до выполнения условия максимальной точности.

Перед пуском программы 121 в ЗУПВ вводят коэффициенты нормированного уравнения $a_2 = P2$, $a_1 = P3$, $a_0 = P4$. Вычисленное значение корня выводится на индикатор и заносится в регистр 5, а исходные данные сохраняются в своих регистрах. Время счета по программе около 200 с. После выполнения программы 121 в УЗП вводят (при сохранении содержимого регистров памяти) программу 122.

Программа 122. Вычисление корней приведенного квадратного уравнения

F2 ↑ F5 + P8 ↑ F5 \times ↑ F3 + P7
↑ F5 \times ↑ F4 + → F8 2 \div /—/ P6
 x^2 ↑ F7 — $x < 0$ РВП /—/ 'V P8 6 0 8
0 C/П V↑ F6 XY P6 — P7 F6 + P6
6 7 C/П .

Результаты вычислений по этой программе выводятся следующим образом: если корни x_2 и x_3 вещественны, то на индикаторе высвечиваются цифры 67, указывающие номера регистров ЗУПВ, в которых хранятся значения корней; если корни комплексные, то высвечиваются цифры 6080, указывающие, что вещественная и мнимая части этой пары корней занесены соответственно в регистры 6 и 8. Значение вещественного корня x_1 сохраняется в регистре 5.

При решении уравнения $x^3 + 1000x^2 + x + 1 = 0$ по программе 121 ($1000 = P2$, $1 = P3$, $1 = P4$) на индикаторе высвечивается значение вещественного корня $x_1 = -999,999$. После ввода

и выполнения программы 122 высвечиваются цифры 6080 и содержимое регистров $P6 = -0,005$; $P7 = 0,0008660254$. Следовательно, $x_{2,3} = -0,0005 \pm j0,0008660254$.

Для решения нормированного уравнения четвертой степени

$$x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0$$

воспользуемся методом Феррари, реализуя его последовательным выполнением программ 123 — 125, 121. Первой используется программа 123 ($a_3 = P5$, $a_2 = P6$, $a_1 = P7$, $a_0 = P8$), формирующая вспомогательное кубическое уравнение $y^3 + b_2y^2 + b_1y + b_0 = 0$ с коэффициентами $b_2 = -a_2/2$; $b_1 = a_1a_3/4 - a_0$; $b_0 = ((4a_2 - a_3^2)a_0 - a_1^2)/8$. После выполнения этой программы содержимое регистров 5 — 8 сохраняется, а коэффициенты b_2 , b_1 и b_0 вспомогательного кубического многочлена хранятся в регистрах 2 — 4 соответственно. На индикаторе высвечиваются цифры 21, указывающие, что следующей выполняют программу 121. После ввода (без изменения содержимого регистров памяти) и выполнения программы 121 вводят и пускают программу 124, вычисляющую коэффициенты квадратных уравнений, на которые разделяется исходное уравнение четвертой степени

$$x^2 + c_1x + c_0 = 0 \text{ и } x^2 + d_1x + d_0 = 0,$$

где $c_1 = (a_3 + D)/2$; $c_0 = y + (a_3y - a_1)/D$; $d_1 = (a_3 - D)/2$; $d_0 = y - (a_3y - a_1)/D$; $D = \sqrt{8y - (4a_2 - a_3^2)}$.

Программа 123. Формирование вспомогательного уравнения

F7	→	F5	→	0	→	F7	→	F5	↑	→	F7
×		÷	↑	F8	—	P3	F6	4	×	↑	F5
x^2	—	↑	→	F8	×	↑	F7	x^2	—	8	÷
P4	F6	2	÷	/—/ P2	2	1	C/Π				

Программа 124. Вычисление коэффициентов квадратичных множителей

F5	8	×	↑	←	—	$x < 0$	5	/—/	$\sqrt{-}$	P6	2
÷	P3	←	↑	2	÷	P2	F5	P4	×	↑	←
—	↑	F6	÷	/—/ P5	2	0	C/Π	$\sqrt{-}$	P6	←	
↑	F6	+	2	÷	←	P7	F5	×	↑	F7	—
↑	F6	÷	←	F6	/—/ $x \geq 0$	F/—/ 2	5	C/Π			

Программа 125. Вычисление корней квадратичных множителей

←	←	2	÷	/—/ P4	x^2	↑	←	—	$x < 0$	F÷
/—/	$\sqrt{-}$	P5	4	0	5 0	C/Π	BП	P0	$\sqrt{-}$	P7
F4	↑	F7	+	P2	F7	—	P3	0	2	0 3
C/Π	BП	P0								

Если коэффициенты квадратных уравнений вещественны, то они заносятся по программе 124 в стек ($C2 = c_1$, $C3 = c_0$, $C5 = d_1$, $C6 = d_0$), а на индикаторе высвечиваются цифры 25, указывающие,

что следующей выполняют программу 125. Если коэффициенты комплексны, то $P2 = Rec_1 = Red_1$; $P3 = Imc_1 = -Imd_1$, $P4 = Rec_0 = Red_0$, $P5 = Imc_0 = -Imd_0$, а на индикаторе высвечиваются цифры 20, указывающие, что следующей выполняется программа 120.

После введения в УЗП программы 125 нажимают клавиши В/О и С/П для вычисления корней первого квадратного уравнения, а затем нажимают клавишу С/П для вычисления корней второго квадратного уравнения. Каждое выполнение программы оканчивается выводом на индикатор цифр 0203 ($P2 = x_1$; $P3 = x_2$) при вещественных корнях или 4050 ($P4 = Rex_{1,2}$, $P5 = Imx_{1,2}$) при комплексно-сопряженных корнях.

При комплексных коэффициентах, вычисленных программой 124, по программе 120 высчитывается одна пара корней, отличающаяся от второй пары только знаком мнимой части.

Например, для решения уравнения четвертой степени

$$x^4 + 4,2679492x^3 + 15,071797x^2 + 23,751288x + 5,358987 = 0$$

заносят в запоминающие устройства программу 123 и коэффициенты многочлена. После выполнения этой программы (время счета около 10 с) на индикаторе высчитываются цифры 21. По программе 121 высчитывают (время счета около 200 с) корень $y = 5,267949$ вспомогательного кубического уравнения, высчитывающийся на индикаторе. После выполнения программы 124 высчитываются цифры 25, а после ввода и выполнения программы 125 — цифры 0203 ($P2 = -0,2679754$; $P3 = -1,999974$). При повторном пуске этой программы нажатием клавиши С/П высчитываются цифры 4050 ($P4 = -0,9999994$; $P5 = 2,999992$). Следовательно, $x_1 = -0,2679754$; $x_2 = -1,999974$; $x_{3,4} = -0,9999994 \pm j2,999992$.

Для решения нормированного уравнения пятой степени

$$x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0$$

первой в УЗП заносят программу 126 ($a_4 = P2$, $a_3 = P3$, $a_2 = P4$, $a_1 = P5$, $a_0 = P6$, $a_0 = P8$). После выполнения этой программы значение вещественного корня хранится в регистре 7 (время счета около 5 мин).

Программа 126. Вычисление вещественного корня уравнения пятой степени

F8	2	\times	P8	/—/	P7	ПП	F/—/	$x < 0$	P0	F8	2
÷	P8	↑	F7	+	P7	÷	sin	$x = 0$	÷	C/П	ПП
F/—/	$x \geq 0$	FXY	F7	↑	F8	—	P7	БП	FXY	F7	↑
F2	ПП	—	F3	ПП	—	F4	ПП	—	F5	ПП	—
F6	XY	+	×	B/O	+	↑	F7	×	↑	B/O	

Затем заносят в УЗП ипускают программу 127, вычисляющую коэффициенты нормированного уравнения четвертой степени ($P5 = a'_3$, $P6 = a'_2$, $P7 = a'_1$, $P8 = a'_0$), получаемого делением исходного уравнения на вычисленный по программе 125 множитель веществен-

ного корня. После выполнения этой программы на индикаторе высвечиваются цифры 23, указывающие номер выполняемой следующей программы 123, а характеризующий точность вычислений остаток от деления хранится в регистре Y.

Программа 127. Деление уравнения пятой степени на множитель вещественного корня

$$\begin{array}{ccccccccc}
 F7 & \uparrow & F2 & + & P2 & \times & \uparrow & F3 & + \\
 \times & \uparrow & F4 & + & P4 & \uparrow & F7 & \times & \uparrow \\
 \uparrow & F7 & \times & \uparrow & F6 & + & P6 & \uparrow & F5 \\
 F3 & P6 & F2 & P5 & XY & 2 & 3 & C/P & P8 & F4 & P7
 \end{array}$$

После выполнения программы 126 при решении уравнения

$$x^5 + 8x^4 + 31x^3 + 80x^2 + 94x + 20 = 0$$

получим $P7 = x_1 = -3,732051$, а после выполнения программы 127 — $P8 = a'_0 = 5,358987$; $P7 = a'_1 = 23,75128$; $P6 = a'_2 = 15,07179$; $P5 = a'_3 = 4,267949$. Вычисления после введения в УЗП программы 123 производятся согласно ранее рассмотренному алгоритму вычисления корней уравнения четвертой степени. Результаты этих вычислений приведены в предыдущем контрольном примере для решения уравнения четвертой степени.

Для решения нормированного уравнения шестой степени

$$x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0$$

можно воспользоваться методом Лобачевского — Греффе, реализуемого программой 128 ($a_5 = C2$, $a_4 = C3$, $a_3 = C4$, $a_2 = C5$, $a_1 = C6$, $a_0 = PX$).

Программа 128. Решение уравнения шестой степени методом Лобачевского—Греффе

$$\begin{array}{ccccccccc}
 \leftarrow & 1 & P5 & 0 & P3 & P4 & \leftarrow & P6 & 0 & \leftarrow & P7 & 0 \\
 \uparrow & \leftarrow & P8 & F3 & P2 & F4 & P3 & F5 & P4 & F6 & P5 & F7 \\
 P6 & F8 & P7 & XY & \leftarrow & P8 & \uparrow & F2 & \times & P2 & F7 & \uparrow \\
 F3 & \times & \uparrow & F2 & — & P2 & F4 & \uparrow & F6 & \times & \uparrow & F2 \\
 — & 2 & \times & /—/ & \uparrow & F5 & x^2 & + & C/P & \uparrow & БП & 2
 \end{array}$$

При использовании программы 128 вычисления целесообразно организовать согласно следующему алгоритму:

1. Составить вычислительный бланк (табл. 52), в начальную строку которого записать коэффициенты заданного уравнения, принять $n = 1$ и перейти к п. 2.

2. Ввести в УЗП программу 128 и в регистры памяти исходные данные и перейти к п. 3.

3. Нажать клавиши В/О и С/П и записать высвечиваемое на индикаторе значение A_{5n} . Нажимая клавишу С/П, выполнить программу и записать в n -ю строку остальные квадрированные коэффициенты A_{in} . Перейти к п. 4.

4. Если очередное значение A_{in} приближается к верхней границе чисел, то прекратить вычисления, иначе принять $n = n + 1$ и перейти к п. 3.

5. Проверить условия разделения коэффициентов, для чего под последней строкой записать строку квадратов коэффициентов предыдущей строки. Определить корни уравнений по формулам (64) согласно сопровождающему их описанию.

Пример 61. Решить уравнение

$$x^6 - 15x^5 + 56x^4 - 56x^3 - 56x^2 + 172x - 120 = 0.$$

Выполнив вычисления по приведенному выше алгоритму (табл. 52), определим, что условия разделения при $n = 5$ выполняются только для коэффициентов a_5 и a_4 . Следовательно, имеются два вещественных корня

$$|x_1| = (9,999995 \cdot 10^{31})^{1/32} \approx 9,999997 \approx 10; |x_2| = \left(\frac{1,85302 \cdot 10^{47}}{9,999995 \cdot 10^{31}} \right)^{\frac{1}{32}} \approx 2,999999 \approx 3.$$

Невыполнение условия разделения для трех смежных коэффициентов a_3 , a_2 и a_1 свидетельствует о наличии четырех корней с одинаковым модулем

$$|x_{3, 4, 5, 6}| = \left(\frac{3,418216 \cdot 10^{66}}{1,85302 \cdot 10^{47}} \right)^{\frac{1}{4-26}} = 1,414213,$$

связанных двумя уравнениями второго порядка. Два из этих корней вещественные и, как показывает их подстановка в исходное уравнение, имеют различные знаки: $x_3 = 1,414213$; $x_4 = -1,414213$.

По уравнению $15 = 10 + 3 + 1,414213 - 1,414213 + 2\operatorname{Re} x_{5, 6}$ находим $\operatorname{Re} x_{5, 6} = 1$ и $\operatorname{Im} x_{5, 6} = \sqrt{1,414213^2 - 1} = 0,9999997 \approx 1$. Следовательно, $x_{5, 6} = 1 \pm j$.

При округлении корней учтена точность разделения, определяемая, согласно данным табл. 52, четырьмя знаками для первых коэффициентов.

Используя программы 14, 15 и 110, можно составить программу автоматического сокращения интервала нахождения вещественного корня уравнений высокого порядка методом половинного деления. Примером может служить программа 129 для нормированного уравнения степени 7 с исходными данными $a_6 = P2$, $a_5 = C1$, $a_4 = C2$, $a_3 = C3$, $a_2 = C4$, $a_1 = C5$, $a_0 = C6$, $x_l = P7$, $\Delta x = P6$, $A = P8$, где x_l — левая граница интервала нахождения корня; Δx — ширина этого интервала; A — любое отличное от нуля число, совпадающее по знаку со значением многочлена при $x = x_l$. Значение корня после выполнения программы высвечивается на индикаторе, а значение невязки заносится в регистр 8 (см. приложение).

<i>n</i>	<i>a₅</i>	<i>a₄</i>	<i>a₃</i>
0	-15	56	-56
1	113	1344	4488
2	10081	809968	-322592
3	$1,000066 \cdot 10^8$	$6,625004 \cdot 10^{11}$	$4,2144 \cdot 10^{13}$
4	$9,999998 \cdot 10^{15}$	$4,304774 \cdot 10^{23}$	$4,408019 \cdot 10^{26}$
5	$9,999995 \cdot 10^{31}$	$1,85302 \cdot 10^{47}$	$4,857547 \cdot 10^{52}$
	$9,999996 \cdot 10^{31}$	$1,853107 \cdot 10^{47}$	$1,943063 \cdot 10^{53}$

Программа 129. Вычисление вещественного корня нормированного уравнения степени 7

F6	2	÷	P6	↑	F7	XY	+	P7	—	<i>x</i> =0	P×
F7	C/P	F7	↑	F2	+	ΠΠ	PC _x	ΠΠ	PC _x	ΠΠ	PC _x
ΠΠ	PC _x	ΠΠ	PC _x	ΠΠ	PC _x	↑	F3	←	F8	XY	P8
×	<i>x</i> <0	P0	F6	/—/	P6	БП	P0	↑	F7	×	↑
F3	↔	P3	+	B/O							

В отличие от программ 121 и 126, автоматически выполняющих как выделение, так и сокращение интервала, содержащего вещественный корень, для применения программы 129 необходимо предварительно выделить этот интервал. Для этого достаточно по величине коэффициентов при высших степенях определить значение аргумента, при котором знак многочлена отличается от знака свободного члена. Например, для уравнения $x^7 + 2x^6 + x^4 + 6x^3 + -2x + 1 = 0$ легко определить границы интервала $[-2; 0]$, в котором содержится по крайней мере один вещественный корень. Занося в УЗП программу 129($2 = P_2$, $0 = C_1$, $1 = C_2$, $6 = C_3$, $0 = C_4$, $2 = C_5$, $1 = C_6$, $-2 = P_7$, $2 = P_6$, $-1 = P_8$) и пуская программу, примерно через 8 мин получим $x_1 = -0,3649643$ при невязке $P_8 = 0$ (т. е. меньше 10^{-8}).

Затраты времени на вычисление вещественных корней уравнений высокой степени часто можно сократить, сочетая метод проб для грубого уточнения корней с автоматическим вычислением значений многочлена по программам 14 или 15 и метод хорд или касательных для заключительного вычисления корня в обычном режиме. При этом для достаточно точного определения корня обычно требуется не более 10—15 обращений к программам 14 или 15.

Например, вещественный корень уравнения

$$f(x) \equiv x^{11} + 5x^{10} + 3x^9 + 8x^7 + 4x^2 + 10 = 0,$$

находящийся в интервале $[-5; 0]$, можно грубо определить, учитывая значение $f(0) = 10$ и последовательно вычисляя по программе 15 $f(-5) = -6484265$; $f(-2,5) = 7549,953$; $f(-3) = 41599$;

a_2	a_1	a_0
-56	172	-120
8960	16144	14400
25919740	2580730	$2,0736 \cdot 10^8$
$1,009407 \cdot 10^{15}$	$1,075609 \cdot 10^{16}$	$4,299816 \cdot 10^6$
$1,692672 \cdot 10^{29}$	$2,888816 \cdot 10^{31}$	$1,848842 \cdot 10^{33}$
$4,775244 \cdot 10^{67}$	$2,08629 \cdot 10^{62}$	$3,418218 \cdot 10^{66}$
$2,865138 \cdot 10^{58}$	$8,345257 \cdot 10^{62}$	$3,418216 \cdot 10^{66}$

$$f(-4) = 131146; \quad f(-4,2) = -37949,2; \quad f(-4,15) = 23078,14; \\ f(-4,17) = 474,6929; \quad f(-4,171) = -716,6913.$$

Воспользуемся методом хорд для вычисления приращения

$$\Delta x = -0,001 \frac{474,6929}{474,6929 + 716,6913} = -0,0003984381.$$

В этом случае $x = -4,1703984$ и $f(x) = 0,726385$. Уточняя переменную в последних знаках, вычисляем $f(-4,170399) = 0,00669$; $f(-4,1703991) = -0,096445$ и окончательно принимаем $x_1 = -4,170399$ с невязкой $\delta_1 = 0,00669$. Разделив заданное уравнение на множитель $x - x_1$ найденного вещественного корня, можно понизить степень уравнения и упростить вычисление остальных вещественных корней (заданное уравнение в рассматриваемом интервале содержит еще два вещественных корня $x_2 = -1,1361214$ с невязкой $\delta_2 = -10^{-6}$ и $x_3 = -1,6941169$ с невязкой $\delta_3 = 3 \cdot 10^{-6}$).

4. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Длину программы и время счета при решении дифференциальных уравнений можно сократить, нормируя переменные для уменьшения числа коэффициентов. Подобный прием целесообразно использовать и при аналитических методах решения, сводящихся к вычислению суммы членов степенного ряда.

Пример 62. Решить дифференциальное уравнение первого порядка

$$\frac{dz}{dt} + \frac{1}{\tau} z = A$$

с начальными условиями $t_0 = 0$, $z_0 = 0$ в интервале аргумента от 0 до $8/\tau$.

Использовав нормировки $x = t/\tau$ и $y = z/\tau A$, представим заданное уравнение в виде $y' = 1 - y$ с начальными условиями $x_0 = 0$, $y_0 = 0$ и интервалом изменения аргумента от 0 до 8.

Последовательно дифференцируя правую часть полученного уравнения, находим $y'' = 1 - y$; $y''' = 1 - y$; $y^{IV} = 1 - y$; ... и формируем степенной ряд

$$y = y_0 + (1 - y_0)(x - x_0) - \frac{(1 - y_0)}{2!}(x - x_0)^2 + \frac{(1 - y_0)}{3!}(x - x_0)^3 - \dots$$

или, с учетом начальных условий,

$$y = x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \dots = \sum_{k=1}^{\infty} (-1)^k \frac{x^k}{k!}.$$

Сумму членов этого ряда для приращений аргумента $\Delta x = x_{i+1} - x_i = P7$, вычисляют по программе 130. Для вычисления y_1 нажимают клавиши В/О и С/П, для вычисления последующих значений y_i нажимают только клавишу С/П. Результат каждого выполнения программы y_i заносится в регистр 4 и выводится на индикатор. Кроме того, в регистр 8 заносится число членов ряда, суммируемых для получения результата по критерию максимальной точности.

Программа 130. Решение дифференциального уравнения методом последовательного дифференцирования

F7	P2	1	/	-	/	P3	0	P4	P8	F8	1	+	P8
F3	/	-	/	↑	F2	×	↑	F8	÷	P3	↑	F4	XY
+	P4	-	x=0	PXY	F4	C/П	F7	↑	F2	+	P2		
БП	P↑												

Результаты решения по этой программе приведены в табл. 53.

Высокая точность решения, полученная в рассмотренном примере, не характеризует погрешности метода степенного ряда и времени вычислений в общем случае. При увеличении аргумента время вы-

Таблица 53

i	x_i	y_i	t, с	k	y_i точно
0	0	0	—	—	0
1	0,4	0,3296799	35	9	0,329680
2	0,8	0,5506710	40	11	0,550671
3	1,2	0,6988057	45	13	0,698806
4	1,6	0,7981033	56	15	0,798104
5	2,0	0,8646645	65	17	0,864665
6	2,4	0,9092819	70	17	0,909282
7	2,8	0,9391897	73	19	0,939190
8	3,2	0,9592376	75	21	0,959238
9	3,6	0,9726761	79	23	0,972676
10	4,0	0,9816839	83	23	0,981684
11	4,4	0,9877224	87	25	0,987723
12	4,8	0,9917703	95	25	0,991770
13	5,2	0,9944834	99	27	0,994483
14	5,6	0,9963018	103	28	0,996302
15	6,0	0,9975207	105	29	0,997521
16	6,4	0,9983370	113	31	0,998338
17	6,8	0,9988769	116	31	0,998886
18	7,2	0,9992560	119	33	0,999253
19	7,6	0,9994917	126	35	0,999500
20	8,0	0,9996560	132	35	0,999665

числений по программе 130 возрастает, а для многих дифференциальных уравнений с аналитической правой частью степенной ряд сходится медленно и в ближайшей окрестности начальной точки. В этих случаях применение метода степенного ряда нецелесообразно и следует использовать итерационные методы численного интегрирования.

Из одноступенчатых методов численного решения дифференциальных уравнений первого порядка наиболее просто программируется метод Эйлера. Однако погрешности этого метода не всегда удается снизить до требуемой величины даже при значительном уменьшении шага интегрирования.

Независимо от алгоритма решения при уменьшении шага интегрирования до некоторого значений погрешности результата уменьшаются, однако при дальнейшем уменьшении шага погрешности начинают возрастать. Это объясняется тем, что полная погрешность результата определяется не только методической ошибкой, но и погрешностями округления на каждом шаге. При шаге $h \rightarrow 0$ методическая погрешность стремится к нулю, но погрешность округления, пропорциональная числу шагов, возрастает. Начиная с некоторого $h < h_{\text{opt}}$ погрешность округления преобладает и суммарная погрешность возрастает. Оптимальное значение шага h_{opt} зависит от вида уравнения и алгоритма его решения.

Метод Эйлера, реализуемый программой 131, в связи с большой методической погрешностью целесообразно использовать лишь для ориентировочного определения результата решения дифференциального уравнения с минимальными затратами времени.

Программа 131. Реализация метода Эйлера

...	↑	F2	×	↑	
F8	+	P8	F2	↑	F3	+	P3	F8	C/П	БП	P0

В незаполненную часть программы 131 вписывают арифметическое выражение для вычисления правой части решаемого уравнения, а перед ее выполнением вводят исходные данные $x_0 = P3$, $h = P2$, $y_0 = P8$. При первом пуске программы нажимают клавиши В/О и С/П, при последующих — только клавишу С/П. В процессе выполнения программы текущее значение x_i заносится в регистр 3. Результат выводится в регистры 8 и X.

Дифференциальное уравнение первого порядка более точно решается методами второго порядка, к которым относятся исправленный и модифицированный методы Эйлера, реализуемые соответственно программами 132 и 133.

Программа 132. Реализация исправленного метода Эйлера

P8	P7	ПП	Р÷	P7	F2	↑	F3	+	P3	ПП	Р÷
↑	F8	+	2	÷	C/П	БП	P0
...	↑	F2	×	↑	F7	+	B/O

Программа 133. Реализация модифицированного метода Эйлера

P8	ПП	1	ПП	1	+	C/П	БП	P0	P7	F2	↑
F3	+	P3	↑	F2	×	↑	F8	+	B/O

В незаполненную часть программы 132 записывают арифметическое выражение, реализующее вычисления согласно формуле (69в) по значениям x_i и y_i , хранящимся соответственно в регистрах 3 и 7. Перед пуском программы вводят исходные данные $h = P2$, $x_0 = P3$, $y_0 = PX$. Программа 133 отличается от предыдущей тем, что вычисления производятся по формуле (69б), а перед ее пуском вводят $h/2 = P2$, $(x_0 - h/2) = P3$.

Дальнейшее увеличение точности результатов решения одноступенчатыми методами связано с повышением их порядка. Метод Рунге — Кутта четвертого порядка, обычно используемый при программировании универсальных ЭВМ, реализуется при вычислениях на микрокалькуляторе программой 134.

Программа 134. Реализация метода Рунге—Кутта четвертого порядка

P7	P8	3	×	P4	ПП	F5	P4	ПП	P,	+	P4
ПП	F5	+	P4	F7	+	P7	ПП	P,	3	÷	C/П
БП	P0	F2	↑	F3	+	P3
...
...	...	↑	F2	×	↑	F8	+	P7	F4	+	B/O

Незаполненную часть текста программы 134 и свободные регистры памяти (P5, P6 и стек) используют для организации вычисления производной по заносимым в регистры 3 и 7 значениям x_i и y_i соответственно. Исходные данные $h/2 = P2$, $x_0 = P3$, $y_0 = PX$.

Если для вычисления производной незаполненных ячеек программной памяти недостаточно, текст программы 134 можно сократить, исключив операторы БП и P0 (в этом случае клавиши B/O и C/П необходимо нажимать для каждого пуска программы) и предварительно записав в регистр 4 значение $3y_0$ (размещение остальных исходных данных сохраняется). Полученная таким образом программа 135 имеет 24 свободные ячейки программной памяти вместо 19 для программы 134.

Программа 135. Упрощенная реализация метода Рунге — Кутта четвертого порядка

P7	P8	ПП	F4	ПП	P ÷	+	P4	ПП	F4	+	P4
F7	+	P7	ПП	P ÷	3	÷	C/П	F2	↑	F3	+
P3
...
...	↑	F2	×	↑	F8	+	P7	F4	+	P4	B/O

Если и при использовании программы 135 свободных ячеек УЗП недостаточно для вычисления производной, то на основании приведенных алгоритмов можно составить программную реализацию метода Рунге — Кутта третьего порядка.

В связи с ограниченной емкостью запоминающих устройств микроСАМУляторов для решения дифференциальных уравнений можно использовать лишь простейшие из двуступенчатых методов. Например, с помощью программы 136 реализуется двуступенчатый метод, основанный на правиле трапеций [15], согласно которому последующее приближение вычисляется по формуле

$$y_{i+1} = y_i + \frac{h}{2} [3f(x_i, y_i) - f(x_{i-1}, y_{i-1})].$$

Программа 136. Реализация простейшего двуступенчатого метода

...	↑
F4	XY	P4	XY	—	+	+	2	÷	↑	F2	×	
↑	F8	+	P8	C/П	F2	↑	F3	+	P3	БП	P0	

В процессе выполнения программы 136 на i -й итерации хранятся $h = P_2$, $x_i = P_3$, $f(x_{i-1}; y_{i-1}) = P_4$, $y_i = P_8$. Регистры 5—7 ЗУПВ и стек можно использовать для хранения коэффициентов при вычислении правой части дифференциального уравнения. Перед выполнением программы 136 по заданным начальным условиям предварительно вычисляют значение функции $f(x_0 - h, y_{-1})$ и y_0 или $f(x_0, y_0)$ и y_1 . В первом случае исходные данные $h = P_2$, $x_0 = P_3$, $f(x_0 - h, y_{-1}) = P_4$, $y_0 = P_8$, во втором — $h = P_2$, $x_0 + h = P_3$, $f(x_0; y_0) = P_4$, $y_1 = P_8$.

Простой алгоритм двуступенчатого варианта метода прогноза и коррекции реализован программой 137.

Программа 137. Реализация варианта метода прогноза и коррекции

F2	↑	F3	+	P3	F4	×	2	×	↑	F8	ПП	
5	↑	F4	+	↑	F2	×	2	÷	↑	F7	ПП	
5	P4	F7	P8	F6	P7	C/П	БП	P0	+	P6	...	
...	B/O

Согласно этому алгоритму последовательно вычисляются:

1. Приближенное значение в $(i + 1)$ -й точке по формуле

$$\tilde{y}_{i+1} = y_{i-1} + 2hf(x_i, y_i).$$

2. Значение первой производной (правой части уравнения) $f(x_{i+1}, \tilde{y}_{i+1})$ по вычисленному значению y_{i+1} .

3. Уточненное значение функции по формуле коррекции

$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1})].$$

4. Уточненное значение производной $f(x_{i+1}, y_{i+1})$.

Точность вычислений можно повысить, если не ограничиваться однократным использованием формулы коррекции, а повторить

вычисления по этой формуле несколько раз для уточнения приближения. С этой целью в программе 138 организован итерационный цикл, в котором вычисления по п.3 повторяются до равенства разности $y_{i+1}^{(k+1)} - y_{i+1}^{(k)}$ машинному нулю. При зацикливании программы (когда вычисления одного из отсчетов не заканчиваются за время вычисления других отсчетов) следует закончить вычисления в обычном режиме и для повторного пуска программы нажать клавиши В/О и С/П.

Программа 138. Реализация метода прогноза и коррекции с итерационным циклом

F2	↑	F3	+	P3	F4	×	2	×	F8	+	
P6	P8	↑	F4	+
2	÷	↑	F2	×	↑	F7	+	↑	F6	XY	P6
—	x=0	F2	F8	P4	F7	P8	F6	P7	C/П	БП	P0

Абсолютную погрешность рассматриваемого метода можно уменьшить, вычисляя уточненное значение функции по формуле

$$y_{i+1}^{(2)} = (4y_{i+1}^{(1)} + y_{i+1}^{(0)})/5.$$

Это соотношение использовано в программе 139 для уточнения значений y_{i+1} , вычисленного по формуле прогноза и однократной коррекции.

Программа 139. Реализация уточненного метода прогноза и коррекции

F2	↑	F3	+	P3	F4	×	2	×	F8	+
ПП	ВП	↑	F4	+	↑	F2	×	2	÷	F7
P8	+	4	×	↑	F6	+	5	÷	P7	ПП
P4	F7	C/П	БП	P0	P6	В/О

В программах 137—139 незаполненная часть текста предназначена для организации вычисления производной по хранящимся в регистрах 3 и 6 текущим значениям переменных x_i и y_i соответственно. Перед выполнением этих программ необходимо предварительно вычислить $f(x_1, y_1)$ и y_1 и ввести исходные данные $h = P2$, $x_1 = P3$, $f(x_1; y_1) = P4$, $y_1 = P7$, $y_0 = P8$ или вычислить $f(x_0; y_0)$ и y_{-1} и ввести $h = P2$, $x_0 = P3$, $f(x_0; y_0) = P4$, $y_0 = P7$, $y_{-1} = P8$. При первом обращении к программам нажимают клавиши В/О и С/П, при последующих — только клавишу С/П.

Пример 63. Решить дифференциальное уравнение $y' = xy^2$ при начальных условиях $x_0 = 1$, $y_0 = -2$, с шагом $h = 0,2$ в интервале $1 < x < 3$.

Для автоматического решения заданного уравнения по программам 131—139 дополнительно вписываем арифметическое выражение для вычисления производной $F3 \uparrow FN x^2 \times$, где $N = 7$ для программ 132—135, $N = 8$ для программ 131 и 136 и $N = 6$ для программ 137—139. Для программ 136—139 используем значение y_1 , вычисленное по программе 134.

Размещение исходных данных в регистрах памяти и результаты вычислений приведены в табл. 54, где для сравнения даны и точные значения резуль-

татов решения. Вычисления по программе 138 для отмеченных звездочкой результатов зацикливаются и их приходится заканчивать в обычном режиме (потактовым выполнением программы при нажатии клавиши ПП).

Сравнение результатов решения дифференциального уравнения с помощью различных программ в приведенном примере подтверждает увеличение точности одноступенчатых методов при повышении их порядка. Однако погрешность результата зависит также от формы заданного уравнения, в связи с чем малая погрешность результата, полученного при использовании метода Рунге — Кутта четвертого порядка для заданного уравнения, еще не свидетельствует о его бесспорных преимуществах во всех случаях. В частности, необходимость четырехкратного вычисления производной при использовании этого метода приводит к увеличению затрат времени, которые могут оказаться чрезмерными при решении дифференциальных уравнений некоторых видов.

Система из двух дифференциальных уравнений первого порядка

$$z' = f_1(x, y, z); \quad y' = f_2(x, y, z)$$

приближенно решается методом Эйлера по программе 140.

Программа 140. Реализация метода Эйлера для решения системы из двух дифференциальных уравнений

...	↑	F2	×	↑	F7	+
P6	↑	F2	×	↑	F8	
+	P8	F2	↑	F3	+	P3	F6	P7	C/П	БП	P0

В начале этой программы записывают арифметическое выражение для вычисления функции $f_2(x, y, z)$ по значениям x, y и z , хранящимся соответственно в регистрах 3, 7 и 8, а во второй незаполненной части программы (после оператора P6) — арифметическое выражение для вычисления $f_1(x, y, z)$. Перед пуском программы вводят исходные данные $h = P2, x_0 = P3, y_0 = P7, z_0 = P8$. Регистр 6 используют для хранения текущих значений $f_2(x, y, z)$, а в остальные регистры записывают коэффициенты решаемых уравнений. После выполнения программы $PX = y_i, P8 = z_i$. При первом обращении к программе нажимают клавиши В/О и С/П, при последующих — только клавишу С/П.

Более точное решение обеспечивает программа 141, реализующая модифицированный метод Эйлера.

Программа 141. Реализация модифицированного метода Эйлера для решения системы из двух дифференциальных уравнений

ПП	F ÷	×	P2	F3	+	P3	ПП	F ÷	÷	P2	↑
F3	+	P3	F7	P8	F5	P6	C/П	БП	P0
...	...	↑	F2	×	↑	F6	+	←
↑	F2	×	↑	F8	+	P7	→	P5	F2	2	B/0

Программа		131	132	133	134	
Исходные данные		P2 P3 P4 P7 P8 PX	0,2 1 — — —2 —	0,2 1 — — —2 —2	0,1 = 0,2/2 0,9 = 1—0,1 — — —2 —2	0,1 = 0,2/2 1 — — —2
<i>i</i>	<i>x_i</i>					
0	1,0	—2	—2	—2	—2	—2
1	1,2	—1,2	—1,4272	—1,4368	—1,389448	—1,389448
2	1,4	—0,8544	—1,059503	—1,069187	—1,020891	—1,020891
3	1,6	—0,6500001	—0,8134975	—0,8212242	—0,7816008	—0,7816008
4	1,8	—0,5148001	—0,6424391	—0,6482242	—0,6175314	—0,6175314
5	2,0	—0,4193932	—0,5193691	—0,5236380	—0,5001754	—0,5001754
6	2,2	—0,3490369	—0,4281723	—0,4313337	—0,4133495	—0,4133495
7	2,4	—0,2954331	—0,3588568	—0,3612230	—0,3473149	—0,3473149
8	2,6	—0,2535384	—0,3050090	—0,3068036	—0,2959271	—0,2959271
9	2,8	—0,2201119	—0,2623801	—0,2637600	—0,2551546	—0,2551546
10	3,0	—0,1929803	—0,2280743	—0,2291496	—0,2222628	—0,2222628
<i>t</i> , с	8	8	8	8	16	16

Исходные данные для этой программы $h/2 = P2$, $x_0 = P3$, $y_0 = P5 = P6$, $z_0 = P7 = P8$. В качестве буферного используется регистр C1 стека. В первую незаполненную часть программы 141 записывают арифметическое выражение для вычисления $f_2(x, y, z)$, во вторую (после оператора \leftarrow) — для вычисления $f_1(x, y, z)$.

При вычислении некоторых функций в соответствии с табл. 40 нельзя использовать стек. В этих случаях можно воспользоваться программой 142, где буферный регистр отсутствует, но число свободных ячеек УЗП сокращается с 19 до 15. В остальном порядок работы с программами 141 и 142 совпадает, причем для первого обращения к этим программам нажимают клавиши B/O и C/P, при последующих — только клавишу C/P.

Программа 142. Реализация решения системы из двух дифференциальных уравнений (без буферных регистров)

ПП	F ÷	×	P2	F3	+	P3	ПП	F ÷	÷	P2	↑
F3	+		P3	F7	P8	F5	P6	C/P	БП	P0	...
F2	×	↑	F6	+	P6	↑	F2	×	↑
F8	+		P7	F5	↑	F6	P5	XY	P6	F2	2
											B/O

Таблица 54

136	137	138	139	—
0,2 1,2 4 — —1,389448 —	0,2 1,2 2,316678 —1,389448 —2 —	0,2 1,2 2,316678 —1,389448 —2 —	0,2 1,2 2,316678 —1,389448 —2 —	Точное решение $y = -\frac{2}{x^2}$
y_i				
—2 —1,389448 —1,094444 —0,8230327 —0,6655821 —0,5347436 —0,4429131 —0,3706297 —0,3148837 —0,2765133 —0,2348237	—2 —1,389448 —0,9964954 —0,7463550 —0,5835029 —0,4719745 —0,3912844 —0,3303057 —0,2827729 —0,2448803 —0,2141434	—2 —1,389448 —1,013869 —0,7740858 * —0,6110119 —0,4948385 —0,4090539 —0,3438642 * —0,2931434 —0,2528933 —0,2204115	—2 —1,389448 —1,011862 —0,7727864 —0,6106276 —0,4949793 —0,4094508 —0,3443655 —0,2936692 —0,2534035 —0,2208875	—2 —1,388888 —1,020408 —0,781250 —0,6172839 —0,5 —0,4132231 —0,3472222 —0,2958579 —0,2551020 —0,2222222
5	8	60	9	—

Если для вычисления функций f_1 и f_2 необходимо большее число свободных ячеек УЗП, то можно использовать программу 143, где в качестве вспомогательных использованы регистры 4 ЗУПВ и С1 стека.

Программа 143. Реализация решения системы из двух дифференциальных уравнений сложной формы

F2	ПП	P ×	F2	2	×	ПП	P ×	P8	F5	P6	C/P
БП	P0	P4	F2	↑	F3	+	P3
↑	F4	×	↑	F8	+	←
...	...	↑	F4	×	↑	F6	+	P8	→	P7	B/O

Для численного решения дифференциального уравнения второго порядка

$$\frac{d^2y}{dx^2} + \varphi_1(x, y) \frac{dy}{dx} + \varphi_2(x, y) = \varphi_3(x)$$

достаточно преобразовать его в систему из двух уравнений первого порядка

$$\begin{aligned}\frac{dz}{dx} &= \varphi_3(x) - \varphi_1(x, y) \cdot z - \varphi_2(x, y) \equiv f_1(x, y, z); \\ \frac{dy}{dx} &= z\end{aligned}$$

и далее использовать одну из программ 140—143.

Пример 64. Решить дифференциальное уравнение второго порядка (уравнение Бесселя)

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

для $n = 1$ при начальных условиях $x_0 = 0,2$; $y_0 = 0,09950083$; $y'_0 = 0,49235$, обеспечивающих точное определение функции Бесселя первого порядка $y = J_1(x)$.

Преобразовав заданное уравнение в систему из двух уравнений

$$\frac{dy}{dx} = z; \quad \frac{dz}{dx} = \frac{n^2 - x^2}{x^2} y - \frac{z}{x} \equiv \left(\frac{n^2 y}{x} - z \right) \frac{1}{x} - y$$

и используя программу 140, дополненную арифметическими выражениями для вычисления производных, получим программу 144. Для более точного решения заданного уравнения на основе программы 141 составим программу 145. Перед пуском программ 144 и 145 в регистр 4 заносят n .

Программа 144. Решение уравнения Бесселя методом Эйлера

F8	↑	F2	×	↑	F7	+	P6	F4	x^2	↑	F7
×	↑	F3	÷	↑	F8	-	↑	F3	÷	↑	F7
-	↑	F2	×	↑	F8	+	P8	F2	↑	F3	+
P3	F6	P7	C/П	БП	P0						

Программа 145. Решение уравнения Бесселя модифицированным методом Эйлера

ПП	F	÷	×	P2	F3	+	P3	ПП	F	÷	÷	P2	↑
F3	+	P3	F7	P8	F5	P6	C/П	БП	P0	F7	↑		
F2	×	↑	F6	+	←	F4	x^2	↑	F5	×	↑		
F3	÷	↑	F7	-	↑	F3	÷	↑	F5	-	↑		
F2	×	↑	F8	+	P7	→	P5	F2	2	B/O			

Результаты вычислений по программам 144 и 145 при шаге $h = 0,05$ и значения $J_1(x)$ с четырьмя верными знаками приведены в табл. 55. Время счета по программам 144 и 145 соответственно около 7 и 15 с.

Для решения системы из трех дифференциальных уравнений (или уравнения третьего порядка) на микрокалькуляторе с емкостью программной памяти 60 шагов практически пригоден только метод Эйлера, реализуемый программой 146.

Программа 146. Реализация метода Эйлера для решения системы из трех дифференциальных уравнений

...	↑	F2	×	↑	F6	+	←	
...	↑	F2	×	↑	F7	+	←	
...	↑	F2	×	↑	F8	+	P8	
F2	↑	F3	+	P3	→	P7	→	P6	C/П	БП	P0		

Таблица 55

i	x	Программа		Точное значение
		144	145	
0	0,2	0,09950083	0,09950083	0,0995
1	0,25	0,1241183	0,1240261	
2	0,3	0,1485515	0,1483200	0,1483
3	0,35	0,1727524	0,1723369	
4	0,4	0,1966750	0,1960318	0,1960
5	0,45	0,2202737	0,2193603	
6	0,5	0,2435039	0,2422790	0,2423
...
26	1,5	0,5710667	0,5579660	0,5579
27	1,55	0,5782249	0,5644506	
28	1,6	0,5843513	0,5699181	0,5699
...
80	3,7	0,05411519	0,05352338	0,0538
81	3,75	0,03208590	0,03291829	
82	3,8	0,01022891	0,01251035	0,0128
83	3,85	-0,01141117	-0,007659523	
84	3,9	-0,03279031	-0,02755118	-0,0272
85	3,95	-0,05386520	-0,04712525	
86	4,0	-0,07459331	-0,06634329	-0,0660

При решении по этой программе системы из трех уравнений

$$\frac{dz}{dt} = f_1(x, y, z, t); \quad \frac{dy}{dt} = f_2(x, y, z, t); \quad \frac{dx}{dt} = f_3(x, y, z, t)$$

исходные данные размещают следующим образом: шаг $h = P2$, независимая переменная $t_0 = P3$, переменные $x_0 = P6$, $y_0 = P7$, $z_0 = P8$. В качестве буферных используют регистры C1 и C2 стека. В незаполненных частях программы записывают арифметические выражения для вычисления производных f_3 , f_2 и f_1 по значениям переменных t , x , y , и z , хранящимся в регистрах 3, 6, 7 и 8 соответственно. После каждого i -го обращения к программе на индикаторе высвечивается значение x_i , а значения t_i^3 , y_i и z_i хранятся в регистрах 3, 7 и 8 соответственно.

Пример 65. Найти приближенное решение уравнения

$$\frac{d^3x}{dt^3} + 6 \cdot 10^3 \frac{d^2x}{dt^2} + 1,1 \cdot 10^7 \frac{dx}{dt} + 6 \cdot 10^9 x = 0$$

при начальных условиях $x(0) = 0$; $x'(0) = 3$; $x''(0) = -11$.

Для нормирования заданного уравнения воспользуемся заменой $\tau = 10^3 t$. Тогда

$$\frac{dx}{dt} = \frac{dx}{d\tau} \frac{d\tau}{dt} = 10^3 \frac{dx}{d\tau}; \quad \frac{d^2x}{dt^2} = 10^6 \frac{d^2x}{d\tau^2}; \quad \frac{d^3x}{dt^3} = 10^9 \frac{d^3x}{d\tau^3}.$$

После подстановки производных по τ и сокращения на 10^9 получаем

$$\frac{d^3x}{d\tau^3} + 6 \frac{d^2x}{d\tau^2} + 11 \frac{dx}{d\tau} + 6x = 0$$

или систему из трех уравнений

$$dz/dt = -6z - 11y - 6x; \quad dy/dt = z; \quad dx/dt = y.$$

В соответствии с программой 146 составляем программу 147 ($h = 0,1 = P2$, $t_0 = 0 = P3$, $x_0 = 0 = P6$, $y_0 = 3 = P7$, $z_0 = -11 = P8$).

Программа 147. Решение системы из трех дифференциальных уравнений

$$\begin{array}{ccccccccc} F7 & \uparrow & F2 & \times & \uparrow & F6 & + & \leftarrow & F8 \\ & \uparrow & F7 & + & \leftarrow & F8 & 6 & \times & \leftarrow \\ & \leftarrow & F6 & 6 & \times & \uparrow & \rightarrow & + & \uparrow \\ F2 & \times & \uparrow & F8 & + & P8 & F2 & \uparrow & F3 \\ P7 & \rightarrow & P6 & C/P & BPI & P0 & & & P3 \end{array}$$

Результаты решения заданного уравнения по программе 147 приведены в табл. 56. Время вычисления каждого отсчета около 9 с. Для сравнения в табл. 56 приведены точные значения результата при аналитическом решении заданного уравнения $x = 2e^{-10^8 t} - e^{-2 \cdot 10^8 t} - e^{-3 \cdot 10^8 t}$.

Таблица 56

i	t_i	τ_i	y_i по программе 147	y_i (точное значение)	i	t_i	τ_i	y_i по программе 147	y_i (точное значение)
0	0	0	0	0	7	0,0007	0,7	0,6645243	0,6241173
1	0,0001	0,1	0,3	0,2501260	8	0,0008	0,8	0,6355142	0,6060433
2	0,0002	0,2	0,49	0,4183297
3	0,0003	0,3	0,603	0,5262552	25	0,0025	2,5	0,1396676	0,1568789
4	0,0004	0,4	0,6625	0,5901168	26	0,0026	2,6	0,1261054	0,1426208
5	0,0005	0,5	0,68523	0,6220517	27	0,0027	2,7	0,1138159	0,1295909
6	0,0006	0,6	0,683089	0,6311303

Решение систем из четырех и более дифференциальных уравнений первого порядка на микрокалькуляторе с емкостью программы памяти 60 шагов при использовании классических методов практически невозможно, но при отображении линейных дифференциальных уравнений в частотную область или замене их разностными уравнениями порядок решаемого дифференциального уравнения можно повысить.

5. АВТОМАТИЗАЦИЯ ЗАДАЧ ИНТЕРПОЛЯЦИИ И АППРОКСИМАЦИИ

Значения функции $f(x)$ между узлами заданной табличной модели определяют линейной интерполяцией по формуле

$$f(x) = f(x_0) + (x - x_0)[f(x_0 + h) - f(x_0)]/h$$

или квадратичной интерполяцией по формуле

$$f(x) = f(x_0) + (x - x_0)[f(x_0 + h) - f(x_0)]/h + (x - x_0)(h - x + x_0)[f(x_0) - f(x_0 - h) + f(x_0 + h) - f(x_0 + 2h)]/(4h^2),$$

где x — заданное значение аргумента, x_0 — ближайшее меньшее значение аргумента в узле таблицы, h — шаг табличной модели.

Для автоматической интерполяции таблично заданных функций по приведенным выражениям можно использовать соответственно программы 148 и 149.

Программа 148. Линейная интерполяция табличной модели

$$\begin{array}{ccccccccc} \uparrow & F4 & \div & P6 & F3 & \uparrow & F2 & - & \uparrow \\ F2 & + & C/P & B\Psi & P0 & & & & \times & \uparrow \end{array}$$

Программа 149. Квадратичная интерполяция табличной модели

$$\begin{array}{ccccccccc} \uparrow & F6 & \div & P7 & 1 & - & \leftarrow & F2 & \uparrow & F3 & - & \uparrow \\ F4 & - & \uparrow & F5 & + & \uparrow & \rightarrow & \times & 4 & \div & \uparrow & F4 \\ + & \uparrow & F3 & - & \uparrow & F7 & \times & \uparrow & F3 & + & C/P & F\Psi \\ P0 & & & & & & & & & & & \end{array}$$

При использовании программы 148 значения функции $f(x_0)$, $f(x_0 + h)$ и шаг таблицы h вводят соответственно в регистры 2, 3 и 4. В индикаторный регистр заносят значение $x - x_0$. При использовании программы 149 значения функции $f(x_0 - x)$, $f(x_0)$, $f(x_0 + h)$, $f(x_0 + 2h)$ и шаг h записывают соответственно в регистры 2—6, а значение $\Delta x = x - x_0$ заносят в регистр индикации. Результаты вычислений выводятся на индикатор.

Например, для $J_0(0,01) = 0,9975 = P2$, $J_0(0,02) = 0,9900 = P3$, $h = 0,01 = P4$, $x - x_0 = 0,005 = PX$ программа 148 вычисляет $J_0(0,015) = 0,99375$ за 3 с. Однако для четырехзначной таблицы функции Бесселя $J_0(x)$ с шагом 0,01 [3], откуда взяты приведенные значения, погрешность линейной интерполяции составит несколько единиц младшего разряда. В этом можно убедиться, вычислив первые разности данной функции. Поэтому целесообразнее использовать квадратичную интерполяцию. Действительно, по программе 149 при $J_0(0) = 1 = P2$, $J_0(0,01) = 0,9975 = P3$, $J_0(0,02) = 0,9900 = P4$, $J_0(0,03) = 0,9776 = P5$, $h = 0,01 = P6$, $x - x_0 = 0,005 = PX$ примерно за 5 с получим $J_0(0,015) = 0,9943687$ с четырьмя верными знаками.

В общем случае задача интерполяции функций по их табличным моделям с n узлами степенным многочленом ($n - 1$)-й степени включает отыскание коэффициентов интерполирующего многочлена с последующим вычислением значений этого многочлена для заданных значений аргумента между узлами табличной модели. Решение первой части этой задачи обеспечивается описываемыми ниже программами.

Программа 150 предназначена для вычисления коэффициентов многочлена второй степени, интерполирующего табличную модель с тремя равноотстоящими узлами (см. рис. 10, а).

Программа 150. Вычисление коэффициентов интерполирующего многочлена второй степени

$$\begin{array}{ccccccccc} F4 & \uparrow & F2 & - & 2 & \div & \uparrow & F5 & \div \\ F4 & + & 2 & \div & \uparrow & F3 & - & \uparrow & F5 & x^2 & \div & P8 \\ \uparrow & F7 & C/P & B\Psi & P0 & & & & & & & \end{array}$$

Исходные данные для этой программы $y_0 = P2$, $y_1 = P3$, $y_2 = P4$, $h = P5$. Результаты заносятся в регистры $P3 = a_0$, $PX = P7 = a_1$, $PY = P8 = a_2$. По этой программе для $y_0 = 0,2$; $y_1 = 0,5$; $y_2 = 0,7$; $h = 0,1$ коэффициенты $a_0 = 0,5$; $a_1 = 2,5$; $a_2 = -5$ вычисляются примерно за 5 с.

Программа 151. Вычисление коэффициентов интерполирующего многочлена третьей степени

F6	2	\times	x^2	P7	\times	2	\div	P8	F4	\uparrow	F3
—	P4	F3	+	P3	F5	\uparrow	F2	—	P5	F2	+
\uparrow	F3	—	8	\div	\rightarrow	F7	\div	P7	F3	\uparrow	\leftarrow
—	2	\div	P2	F5	3	\div	\uparrow	F4	—	8	\div
\rightarrow	F8	\div	P8	F4	\uparrow	\leftarrow	—	\uparrow	F6	\div	C/P

Программа 151 предназначена для вычисления коэффициентов многочлена третьей степени, интерполирующего табличную модель функции с четырьмя равноотстоящими узлами (см. рис. 10, б). Исходные данные $y_0 = P2$, $y_1 = P3$, $y_2 = P4$, $y_3 = P5$, $h = P6$, вывод результатов $P8 = a_3$, $P7 = a_2$, $P2 = a_0$, $PX = a_1$.

Для $y_0 = 0,5$; $y_1 = 2$; $y_2 = 4,5$; $y_3 = 5,25$; $h = 2$ результаты $P2 = 3,296875$; $PX = 1,307291$; $P7 = -0,046876$; $P8 = -0,05729166$ (т. е. $y = 3,296875 + 1,307291x - 0,046875x^2 - 0,05729166x^3$) вычисляются по программе 151 примерно за 25 с.

Программа 152. Вычисление коэффициентов интерполирующего многочлена четвертой степени

F6	x^2	1	2	\times	2	\times	\rightarrow	F6	\div	\rightarrow	F6
x^2	\times	2	\times	\rightarrow	F6	\div	\rightarrow	PП	$/-$	P7	PП
$/-$	$/-$	P8	PП	$/-$	$/-$	P6	PП	$/-$	P5	C/P	F4
2	\times	$/-$	P4	\uparrow	F5	$/-$	P5	—	\rightarrow	F3	2
\times	P3	\uparrow	F2	—	\uparrow	\leftarrow	+	\uparrow	\leftarrow	\div	B/O

Коэффициенты многочлена четвертой степени ($a_0 = y_2$), интерполирующего табличную модель с пятью равноотстоящими узлами (см. рис. 10, в) вычисляются по программе 152 ($y_0 - y_2 = P2$, $y_1 - y_2 = P3$, $y_3 - y_2 = P4$, $y_4 - y_2 = P5$, $h = P6$; $P8 = a_4$, $P7 = a_3$, $PX = P5 = a_2$, $P6 = a_1$). Для табличной модели $y_t = 10; 20; 35; 50; 55$ с шагом $h = 2$ коэффициенты многочлена $y = 35 + 8,125x + 0,05208333x^2 - 0,15625x^3 - 0,01302083x^4$ по программе 152 вычисляются примерно за 25 с.

Коэффициенты интерполирующего многочлена пятой степени для табличной функции с шестью равноотстоящими узлами (см. рис. 10, г) определяются с помощью последовательно выполняемых программ 153—155. Последняя из этих программ предназначена для пересчета коэффициентов многочлена при заданном шаге h .

Первой вводится в УЗП и выполняется программа 153 ($y_0 = P2$, $y_1 = P3$, $y_2 = P4$, $y_3 = P5$, $y_4 = P6$, $y_5 = P7$, $h = P8$). После ее окончания (время счета около 15 с) на индикаторе высвечива-

ются цифры 154, что указывает на необходимость введения в УЗП и пуска (нажатием клавишей В/О и С/П) программы 154. По выполнении этой программы (время счета около 15 с) в регистрах ЗУПВ хранятся вычисленные нормированные коэффициенты многочлена ($P_2 = a_5 h^5$, $P_3 = a_3 h^3$, $P_4 = a_1 h$, $P_5 = a_0$, $P_6 = a_2 h^2$, $P_7 = a_4 h^4$), а на индикаторе высвечиваются цифры 155, указывающие на выполняемую следующей программу. После выполнения программы 155 (время счета около 10 с) на индикаторе высвечиваются цифры 00, а результаты вычислений хранятся в стеке: $C_1 = a_5$, $C_2 = a_4$, $C_3 = a_3$, $C_4 = a_2$, $C_5 = a_1$, $C_6 = a_0$. Например, для $y_0 = 1$; $y_1 = 2$; $y_2 = 4$; $y_3 = 7$; $y_4 = 6$; $y_5 = 4,5$; $h = 2$ по программам 153—155 вычисляются коэффициенты интерполирующего многочлена $y = 5,728515 + 1,635807x - 0,2330729x^2 - 0,1393229x^3 + 0,004557291x^4 + 0,003515625x^5$.

Программа 153. Подготовка данных для вычисления интерполирующего многочлена пятой степени

$$\begin{array}{ccccccccc}
 F7 & \uparrow & F2 & + & P7 & F2 & - & P2 & F6 & \uparrow & F3 & + \\
 P6 & F3 & - & P3 & F5 & \uparrow & F4 & + & P5 & F4 & - & P4 \\
 2 & \times & \uparrow & F3 & - & 5 & \times & \uparrow & F2 & + & 1 & 2 \\
 0 & \div & P2 & F5 & 2 & \times & 3 & \div & \uparrow & F6 & - & 3 \\
 \times & \uparrow & F7 & + & 4 & 8 & \div & P7 & 1 & 5 & 4 & C/P
 \end{array}$$

Программа 154. Вычисление нормированных коэффициентов интерполирующего многочлена пятой степени

$$\begin{array}{ccccccccc}
 F2 & 5 & \times & \uparrow & F4 & + & 3 & \times & /-/ \\
 6 & \div & P3 & 4 & \times & \uparrow & F2 & + & 1 \\
 \uparrow & F4 & + & P4 & F7 & 1 & 0 & \times & \uparrow \\
 \uparrow & F6 & + & 4 & \div & P6 & F7 & + & 8 \\
 F5 & + & 2 & \div & P5 & 1 & 5 & 5 & C/P
 \end{array}$$

Программа 155. Вычисление коэффициентов интерполирующего многочлена пятой степени

$$\begin{array}{ccccccccc}
 F5 & \rightarrow & F4 & \uparrow & F8 & \div & \rightarrow & F6 & \uparrow & F8 & x^2 & \div \\
 \rightarrow & F3 & \uparrow & F8 & x^2 & \div & \uparrow & F8 & \div & \rightarrow & F7 & \uparrow \\
 F8 & x^2 & x^2 & \div & \rightarrow & F2 & \uparrow & F8 & x^2 & x^2 & \div & \uparrow \\
 F8 & \div & \rightarrow & 0 & 0 & C/P
 \end{array}$$

Многочлены, коэффициенты которых вычисляются по программам 150—155, соответствуют началу отсчета аргумента в средине интервала его изменения (см. рис. 12). Для переноса начала отсчета аргумента в точку $x = x_0$ интерполирующего многочлена степени $m \leq 5$ предназначена программа 156 ($a_0 = C_6$, $a_1 = C_5$, $a_2 = C_4$, $a_3 = C_3$, $a_4 = C_2$, $a_5 = C_1$, $x_0 = P_2$). Если $m < 5$, то на место равных нулю коэффициентов заносят нули. При использовании программы 156 после программы 155 исходные данные дополнительно вводить не нужно.

Программа 156. Перенос начала отсчета интерполирующего многочлена степени $m \leq 5$

1	P5	6	P4	ПП	F4	ПП	3	ПП	P \div	ПП	F3
ПП	P3	\rightarrow	\rightarrow	\rightarrow	C/П	\leftarrow	\leftarrow	\leftarrow	F5	1	+
P5	F4	1	—	P4	P3	\leftarrow	P8	P7	F3	\uparrow	F7
\times	\uparrow	F5	\div	\leftarrow	P7	F8	\uparrow	F2	\times	\uparrow	F7
+	P8	F3	1	—	P3	x = 0	5	F8	\leftarrow	B/O	

После выполнения программы 156 (время счета около 70 с) коэффициенты преобразованного многочлена оказываются занесенными в регистры стека на места исходных коэффициентов. Это позволяет повторять смещение начала отсчета без перезаписи коэффициентов. При каждом обращении к программе 156 нажимают клавиши B/O и C/П.

Например, при $a_0 = 5,728515$; $a_1 = 3,271614$; $a_2 = -0,9322917$; $a_3 = -1,114583$; $a_4 = 0,07291666$; $a_5 = 0,1125$; $x_0 = -1,5$ по программе 156 вычисляем $a'_0 = 1,999999$; $a'_1 = 0,408335$; $a'_2 = 1,270831$; $a'_3 = 0,9792671$; $a'_4 = -0,7708333$; $a'_5 = 0,1125$, а повторное выполнение программы при предварительном вводе $x_0 = 3$ дает $a''_0 = 5,999998$; $a''_1 = -3,216663$; $a''_2 = -1,166663$; $a''_3 = 1,854167$; $a''_4 = 0,916667$; $a''_5 = 0,1125$.

Если начало отсчета аргумента совпадает с первым узлом табличной модели, а $m \leq 5$, то коэффициенты интерполирующих многочленов удобнее вычислять по программам 157—159.

Программа 157. Вычисление коэффициентов интерполирующего многочлена при единичном шаге

P2	\uparrow	ПП	F6	\uparrow	P3	ПП	FBП	\leftarrow	2	\div	P4
ПП	F7	\leftarrow	\leftarrow	6	\div	P5	ПП	FC _x	\leftarrow	\leftarrow	\leftarrow
2	4	\div	P6	\leftarrow	XY	—	1	2	0	\div	P7
C/П	\leftarrow	XY	—	\leftarrow	XY	—	\leftarrow	XY	—	\leftarrow	XY
—	\leftarrow	XY	—	\leftarrow	\leftarrow	\leftarrow	\leftarrow	B/O			

Программа 158. Вычисление нормированных коэффициентов интерполирующего многочлена степени $m \leq 5$

F7	2	5	\times	P8	F7	1	0	\times	\uparrow	F8	—
\rightarrow	F6	XY	—	P6	6	\times	\uparrow	F8	$+$	\uparrow	F5
XY	—	P5	3	\times	\rightarrow	F6	7	\times	$/-$	\uparrow	\leftarrow
—	\uparrow	\leftarrow	$+$	\uparrow	F4	$+$	P4	F3	\uparrow	F4	—
\uparrow	F5	—	\uparrow	F6	—	\uparrow	F7	—	P3	C/П	

Программа 159. Вычисление коэффициентов интерполирующего многочлена Ньютона степени $m \leq 5$

F2	\rightarrow	F3	\uparrow	F8	\div	\rightarrow	F4	\uparrow	F8	x^2	\div
\rightarrow	F5	\uparrow	F8	x^2	\div	\uparrow	F8	\div	\rightarrow	F6	\uparrow
F8	x^2	x^2	\div	\rightarrow	F7	\uparrow	F8	x^2	x^2	\div	
F8	\div	\rightarrow	0	C/П							

Программа 157 обеспечивает вычисление коэффициентов интерполирующего многочлена Ньютона (17а) при шаге $h = 1$, а программа 158— пересчет коэффициентов по формулам

$$a_5 = \frac{\Delta^5(0)}{5!}; \quad a_4 = \frac{\Delta^4(0)}{4!} - 10a_5; \quad a_3 = \frac{\Delta^3(0)}{3!} - 6a_4 - 25a_5;$$

$$a_2 = \frac{\Delta^2(0)}{2!} - 3a_3 - 7a_4 - 15a_5; \quad a_1 = \frac{\Delta^1(0)}{1!} - a_2 - a_3 - a_4 - a_5;$$

$$a_0 = \frac{\Delta^0(0)}{0!} = y(0),$$

которые остаются справедливыми и при $m < 5$. Программа 159 предназначена для случая, когда шаг интерполяции отличается от единицы. Вычисленные по программам 157—159 коэффициенты пересчитывают для другого выбора начала отсчета аргумента с помощью программы 156.

Перед пуском программы 157 вводят исходные данные: $y_1 = C1$, $y_2 = C2$, $y_3 = C3$, $y_4 = C4$, $y_5 = C5$, $y_0 = PX$. После выполнения этой программы (время счета около 20 с) вычисленные коэффициенты многочлена Ньютона размещаются в ЗУПВ: $P2 = \Delta^0(0)/0!$; $P3 = \Delta^1(0)/1!$; $P4 = \Delta^2(0)/2!$; $P5 = \Delta^3(0)/3!$; $P6 = \Delta^4(0)/4!$; $P7 = PX = \Delta^5(0)/5!$. Последующие ввод и выполнение программы 158 (время счета около 10 с) приводят к результатам: $P2 = a_0$, $P3 = a_1h$, $P4 = a_2h^2$, $P5 = a_3h^3$, $P6 = a_4h^4$, $P7 = a_5h^5$, $P8 = PX = a_1h$. Затем в регистр 8 вводят расстояние h между узлами модели и в УЗП заносится программа 159. Результат ее выполнения по хранимым в памяти ранее полученным данным: $C1 = a_5$, $C2 = a_4$, $C3 = a_3$, $C4 = a_2$, $C5 = a_1$, $C6 = a_0$.

Например, для табличной модели $y_i = 2; 4; 6,5; 7; 6; 3$ с шагом $h = 2$ по программе 157 получаем $P2 = 2$; $P3 = 2$; $P4 = 0,25$; $P5 = -0,4166666$; $P6 = 0,125$; $P7 = -0,03333333$; по программе 158 получаем $P2 = 2$; $P3 = -0,63333333$; $P4 = 4,541666$; $P5 = -2,333333$; $P6 = 0,45833333$; $P7 = -0,03333333$. Вводя $h = 2 = P8$ и выполняя программу 159, получаем $C1 = -0,001041666$; $C2 = 0,02864583$; $C3 = -0,29166666$; $C4 = 1,135416$; $C5 = -0,31666666$; $C6 = 2$. Следовательно, $y = 2 - 0,31666666x + 1,135416x^2 - 0,29166666x^3 + 0,02864583x^4 - 0,001041666x^5$.

Для линейной аппроксимации методом наименьших квадратов предназначена программа 160, вычисляющая коэффициенты a_0 и a_1 многочлена первой степени, аппроксимирующего функциональную зависимость, заданную дискретным множеством точек с координатами y_i и x_i . По этой программе автоматически формируется и решается относительно искомых коэффициентов система уравнений (28).

Программа 160. Линейная аппроксимация методом наименьших квадратов

P2	F5	+	P5	F2	×	↑	F6	+	P6	F2	↑
F7	+	P7	F2	×	↑	F8	+	P8	F4	1	+
P4	C/P	F8	↑	F7	÷	↑	F5	×	P2	F4	×
↑	F7	-	P3	F2	↑	F6	-	↑	F3	÷	P2
↑	F4	×	↑	F5	XY	-	↑	F7	÷	P3	C/P

Перед пуском программы 160 обязательно очищают регистры 4—8 и в регистры Y и X соответственно последовательно вводят координаты y_i и x_i с нажатием клавиш В/О и С/П после каждого ввода (на индикаторе высвечивается число обработанных точек). После обработки всех точек нажимают клавишу С/П и результаты вычислений по выполненной программе размещаются в регистрах $P2 = a_0$, $P4 = n + 1$, $P5 = \sum y_i$, $P6 = \sum y_i x_i$, $P7 = \sum x_i$, $P8 = \sum x_i^2$, $P3 = PX = a_1$.

Работу с программой 160 можно иллюстрировать на примере аппроксимации табличной модели $(y_i, x_i) = (0; -2,5), (1; -2), (2; -1), (2; 0), (3; 1), (4, 5; 2), (5; 3), (5; 4), (6; 5)$. После занесения программы в УЗП выполняемые пользователем операции в этом случае можно описать предложением

0	P4	P5	P6 - P7	P8	0	↑	2,5	/ - /	B/O	C/P	(1)		
1	↑	2	/ - /	B/O	C/P	(2)	2	↑	1	/ - /	B/O	C/P	
(3)	2	↑	0	B/O	C/P	(4)	3	↑	1	B/O	C/P	(5) 4,5	
↑	2	B/O	C/P	(6)	5	↑	3	B/O	C/P	(7)	5	↑	4
B/O	C/P	(8)	6	↑	5	B/O	C/P	(9)	C/P	(0,7633401)	F2	(2,360918).	

Следовательно, $y = 2,360918 + 0,7633401x$.

В общем случае задача определения коэффициентов аппроксимирующего многочлена m -й степени методом наименьших квадратов заключается в формировании и решении системы (28) из $m + 1$ уравнений при $n > m + 1$ узлах аппроксимируемой табличной модели. В связи с ограниченной емкостью запоминающих устройств программируемых микрокалькуляторов решение этой задачи при $m \geq 2$ с помощью одной или даже двух программ затруднительно. Однако подобную аппроксимацию можно упростить при выборе начала отсчета аргумента в середине интервала так, чтобы точки отсчета функции были попарно симметричны относительно этого начала. Это требование легко выполнить для табличной модели с четным или нечетным числом равноотстоящих узлов.

При выполнении условия парной симметрии узлов отсчета коэффициенты c_k с нечетными индексами в системе уравнений (28) равны нулю и система распадается на две независимых, которые можно решать раздельно. Для аппроксимации табличной модели с $n + 1$ узлами степень многочлена можно выбирать любой в пределах от

1 до $n - 1$. Коэффициенты аппроксимирующих многочленов первой, второй и третьей степени вычисляют с помощью программ 161—165.

В случае линейной аппроксимации многочленом вида $a_0 + a_1x$ система уравнений (28) при парной симметрии узлов отсчета относительно середины интервала изменения функции вырождается в матричное уравнение

$$\begin{bmatrix} c_0 & 0 \\ 0 & c_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \end{bmatrix},$$

формируемое и решаемое относительно коэффициентов a_0 и a_1 по программе 161. При работе с этой программой регистры 3—6 предварительно очищают, а в регистры Y и X соответственно поочередно вводят пары значений y_i и x_i с нажатием после каждого ввода клавиш В/О и С/П (на индикаторе после каждого выполнения программы высвечивается число обработанных точек отсчета). Введя последнюю пару значений (порядок ввода не существен), нажимают клавишу С/П. После выполнения программы значение a_1 хранится в регистрах 8 и X, а значение a_0 — в регистре 7.

Программа 161. Линейная аппроксимация методом наименьших квадратов при парной симметрии узлов отсчета

P2	F3	+	P3	F2	\times	↑	F4	+	P4	F2	x^2
↑	F6	+	P6	F5	1	+	P5	C/P	F3	↑	F5
÷	P7	F4	↑	F6	÷	P8	C/P				

Результаты вычислений по программе 161 можно уточнить вводом и обработкой дополнительных отсчетов с парно-симметричными значениями аргументов. В этом случае необходимо сохранить в регистрах 3—6 результаты предыдущих вычислений и обработать дополнительные отсчеты ранее описанным способом. Например, для аппроксимации функции, заданной табличной моделью $(y_i, x_i) = (2,5; -2), (3,7; -1), (4,9; 0), (6; 1), (6,4; 2)$, следует ввести в УЗП программу 161 и, перейдя в рабочий режим, ввести предложение

0	P3	P4	P5	P6	2,5	↑	2	/—/	B/O	C/P	3,7	↑
1	/—/	B/O	C/P	4,9	↑	0	B/O	C/P	6	↑	1	B/O
	C/P	6,4	↑	2	B/O	C/P	C/P	(1,01)	F7	(4,7).		

Следовательно, $y(x) \approx 4,7 + 1,01x$. Для уточнения результатов аппроксимации дополнительными экспериментальными данными $(y_i, x_i) = (-2,9; -8), (0,1; -5), (9, 95; 5), (13, 1; 8)$ следует, не изменяя содержимого регистров памяти, ввести предложение

2,9	/—/	↑	8	/—/	B/O	C/P	0,1	↑	5	/—/	B/O	C/P
9,95	↑	5	B/O	C/P	13,1	↑	8	B/O	C/P	C/P		
					(0,9965425)	F7	(4,861111).					

Следовательно, уточненный аппроксимирующий многочлен $y(x) = 4,861111 + 0,9965425x$.

При парной симметрии узлов табличной модели системы уравнений (28) для аппроксимации квадратичным трехчленом принимает вид

$$\begin{bmatrix} c_0 & 0 & c_2 \\ 0 & c_2 & 0 \\ c_2 & 0 & c_4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix},$$

а ее решение: $a_0 = (q_0c_4 - q_2c_2)/(c_0c_4 - c_2^2)$; $a_2 = (q_2c_0 - q_0c_2)/(c_0c_4 - c_2^2)$; $a_1 = q_1/c_2$. Квадратичная аппроксимация степенным многочленом по методу наименьших квадратов в этом случае реализуется программой 162, формирующей систему уравнений, и программой 163, вычисляющей коэффициенты аппроксимирующего многочлена второй степени.

Программа 162. Формирование системы уравнений для вычисления коэффициентов аппроксимирующего многочлена

$$\begin{array}{ccccccccc} P2 & F3 & + & P3 & F2 & \times & \uparrow & F4 & + \\ \uparrow & F5 & + & P5 & F2 & x^2 & \uparrow & F6 & + \\ \uparrow & F7 & + & P7 & F8 & 1 & + & P8 & C/P \end{array} \quad \begin{array}{ccccc} P4 & F2 & \times & X Y & x^2 \\ P6 & X Y & \times & P0 & \end{array}$$

Программа 163. Вычисление коэффициентов аппроксимирующего многочлена второй степени

$$\begin{array}{ccccccccc} F7 & \uparrow & F3 & \times & \rightarrow & F8 & \times & \uparrow & F6 & x^2 \\ F5 & \uparrow & F6 & \times & \rightarrow & F8 & \times & \rightarrow & F6 & \uparrow \\ \rightarrow & F4 & X Y & \div & P7 & \rightarrow & \rightarrow & \Pi \Pi & RBP \\ P8 & C/P & \rightarrow & \uparrow & \rightarrow & - & \uparrow & F2 & \div \\ & & & & & & & B/O & \end{array} \quad \begin{array}{ccccc} P2 & \times & F3 & \Pi \Pi & RBP \\ P6 & \Pi \Pi & RBP & & \end{array}$$

При использовании этих программ первой в УЗП вводят программу 162 и очищают регистры 3—8. После этого в регистры Y и X соответственно вводят пары значений y_i и x_i с нажатием клавиши B/O и C/P для первого пуска и клавиши C/P для последующих пусков программы. При этом очередную пару значений вводят после высвечивания на индикаторе числа ранее введенных пар значений y_i и x_i . После обработки исходных данных по программе 162 результаты вычислений хранятся в регистрах ЗУПВ: P3 = q_0 , P4 = $= q_1$, P5 = q_2 , P6 = c_2 , P7 = c_4 , P8 = PX = $c_0 = n$. Не изменяя содержимого регистров памяти, в УЗП вводят программу 163 и нажимают клавиши B/O и C/P. После выполнения этой программы P3 = q_0 , P4 = q_1 , P5 = q_2 , P6 = a_0 , P7 = a_1 , P8 = PX = a_2 .

Например, при квадратичной аппроксимации табличной модели $(y_i; x_i) = (-17; -5), (-6; -3), (-0,375; -1,5), (4,125; 1,5), (3; 3), (-2; 5)$ после выполнения программы 162 получаем P3 = $-18,25$; P4 = $108,75$; P5 = $-493,5625$; P6 = $72,5$; P7 = $1422,125$; P8 = PX = 6 , и после выполнения программы 163 получаем P6 =

$= 3; P7 = 1,5; P8 = PX = -0,5$. Следовательно, $y \approx 3 + 1,5x - 0,5x^2$.

Вычисление коэффициентов аппроксимирующего многочлена третьей степени при парной симметрии узлов отсчета реализуется с помощью программ 164 и 165.

Программа 164. Формирование системы уравнений для вычисления коэффициентов аппроксимирующего кубического многочлена

P2	F3	+	P3	F2	\times	\uparrow	F4	+	P4	F2	\times
\uparrow	F5	+	P5	F2	\times	\uparrow	F6	+	P6	F2	x^2
\uparrow	F7	+	P7	XY	x^2	\uparrow	F8	+	P8	XY	\uparrow
$\sqrt{ }$	\times	\uparrow	\rightarrow	$+$	\leftarrow	\leftarrow	1	$+$	\uparrow	\rightarrow	F1
C/P	P2	F8	\leftarrow	F7	\leftarrow	F6	\leftarrow	F4	\leftarrow	\leftarrow	C/P

Программа 165. Вычисление коэффициентов аппроксимирующего многочлена третьей степени

PП	P ÷	P3	\rightarrow	P5	\rightarrow	P2	\rightarrow	P7	\rightarrow	P8	\leftarrow
PП	P ÷	\rightarrow	\rightarrow	0	0	0	C/P	F2	\uparrow	F8	\times
\uparrow	F7	x^2	$-$	P4	F5	\uparrow	F2	\times	P2	F7	\times
P6	F3	\uparrow	F7	\times	P5	F8	\times	\uparrow	F6	$-$	\uparrow
F4	\div	\rightarrow	F2	\uparrow	F5	$-$	\uparrow	F4	\div	\rightarrow	B/O

Первая из этих программ формирует систему уравнений

$$\begin{bmatrix} c_0 & 0 & c_2 & 0 \\ 0 & c_2 & 0 & c_4 \\ c_2 & 0 & c_4 & 0 \\ 0 & c_4 & 0 & c_6 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix},$$

а вторая вычисляет коэффициенты аппроксимирующего многочлена

$$a_0 = (q_0c_4 - q_2c_2)/(c_0c_4 - c_2^2); \quad a_1 = (q_1c_6 - q_3c_4)/(c_2c_6 - c_4^2);$$

$$a_2 = (q_2c_0 - q_0c_2)/(c_0c_4 - c_2^2); \quad a_3 = (q_3c_2 - q_1c_4)/(c_2c_6 - c_4^2).$$

После введения в УЗП программы 164 очищают регистры 3—8 и в регистры Y и X соответственно вводят пары значений y_i и x_i с нажатием клавиши B/O и C/P после каждого ввода (очередная пара значений вводится после высвечивания на индикаторе числа ранее введенных пар значений). Окончив обработку всех отсчетов, нажимают клавишу C/P. Результаты вычисления хранятся в регистрах памяти $P2 = c_0$, $P3 = q_0$, $P4 = q_1$, $P5 = q_2$, $P6 = q_3$, $P7 = c_2$, $P8 = c_4$, $C1 = c_6$, $C2 = c_4$, $C3 = c_2$, $C4 = q_3$, $C5 = q_1$, $C6 = 0$. После выполнения занесенной в УЗП программы 165 на индикаторе высвечиваются цифры 000 и искомые коэффициенты находятся в стеке: $C1 = a_2$, $C2 = a_0$, $C3 = a_3$, $C4 = a_1$.

Например, при аппроксимации кубическим трехчленом табличной модели $(y_i, x_i) = (45; -4), (9; -2), (4,5; -1), (5; 0), (7,5; 1), (9; 2), (3; 4)$ по программе 164 получаем $P2 = 7; P3 = 83; P4 = -165; P5 = 852; P6 = -2625; P7 = 42; P8 = 546$, и после

выполнения программы 165 получаем $C1 = 1,204081$; $C2 = 4,632683$; $C3 = -0,4411764$; $C4 = 1,806722$ или $y = 4,632653 + 1,806722x + + 1,204081x^2 - 0,4411764x^3$.

Необходимое для применения программ 161—165 требование парной симметрии узлов отсчета не ограничивает общность метода, так как при нечетном числе узлов модели в качестве середины интервала можно принять центральный узел, а при четном — середину интервала между двумя узлами, расположенными в центре интервала изменения аргумента.

Наряду со степенными многочленами в качестве интерполирующих или аппроксимирующих функций используют также тригонометрические ряды.

6. АВТОМАТИЗАЦИЯ СПЕКТРАЛЬНОГО АНАЛИЗА

Гармонический анализ на ЭВМ часто выполняют с помощью алгоритмов быстрого преобразования Фурье, которые, однако, малопригодны для программирования микрокалькуляторов с небольшой емкостью запоминающих устройств. Поэтому приходится использовать другие методы вычисления коэффициентов тригонометрических рядов (30).

Последовательный анализ, при котором за каждый цикл вычислений определяется одна пара коэффициентов a_k и b_k тригонометрического ряда, реализуется программой 166.

Программа 166. Вычисление коэффициентов ряда Фурье

P2	0	P5	P7	P8	π	2	\times	\uparrow	F3	\div	\uparrow	F8
F4	\times	P4	F5	\uparrow	F4	\times	\cos	\uparrow	F2	\times	\uparrow	C/P
F7	+	P7	F5	\uparrow	F4	\times	e^{ix}	F2	\times	\uparrow	\uparrow	
+	P8	F5	1	$+$	P5	\uparrow	F3	$-$	$x \neq 0$	8		
P2	БП	2	F7	x^2	\uparrow	F8	x^2	$+$	V	C/P		

После ввода в УЗП этой программы в регистр 3 вводится число n разбиений периода (интервала определения) функции, а в регистр 4 — номер k вычисляемых коэффициентов. В индикаторный регистр последовательно вводятся отсчеты y_0, y_1, \dots, y_n с нажатием клавиши С/П (первый раз В/О и С/П) после высвечивания числа подлежащих вводу отсчетов со знаком минус. При последнем выполнении программы в регистры заносятся нормированные коэффициенты: $P7 = n a_k / 2$, $P8 = n b_k / 2$, $PX = n c_k / 2$. Для вычисления коэффициентов с другими номерами все операции повторяются.

Если вычислены все коэффициенты ряда (104), то он будет интерполирован заданную табличную модель функции, обеспечивая совпадение суммы членов ряда со значениями y_i в узлах модели.

Программу 166 можно также использовать для приближенного определения спектральной плотности непериодических функций, заданных на интервале $\pm\infty$. В этом случае при выбранных интер-

вале дискретизации Δt и числе k величину $n_{ck}/2$ можно рассматривать как деленный на интервал дискретизации модуль спектральной плотности $|S(j\omega)|/\Delta t$ на частоте $\omega_k = 2\pi k/\Delta t$. Аргумент функции $S(j\omega)$ можно определить по ее вещественной и мнимой частям, занесенным программой соответственно в регистры 7 и 8.

Пример 66. Вычислить модуль спектральной плотности прямоугольного импульса

$$y(t) = \begin{cases} 1 & \text{при } 0 \leq t < \tau; \\ 0 & \text{при } t > \tau \end{cases} \quad \text{при } \tau = 1 \text{ мкс}$$

на частотах до $f_{\max} = 1,25 \text{ МГц}$ через 250 кГц .

В результате дискретизации сплошного спектра непериодической функции она периодически повторяется с квазипериодом $T = 1/\Delta f$, где Δf — частотный интервал между смежными значениями функции $S(j\omega)$, подлежащими определению. В рассматриваемой задаче $T = 1/250 \cdot 10^3 = 4 \cdot 10^{-6} = 4 \text{ мкс}$.

Выбор интервала дискретизации Δt определяется требуемой точностью вычислений и должен удовлетворять условию $\Delta t \times 1/f_{\max} = 0,8 \text{ мкс}$. Приняв $\Delta t = 0,125 \text{ мкс}$, определим количество интервалов разбиения квазипериода анализируемой функции $n = T/\Delta t = 4/0,125 = 32$. На этом интервале заданная функция отображается восьмью единичными отсчетами и равными нулю остальными.

Введя в микрокалькулятор программу 166 и записав в регистры 3 и 4 значения $n = 32$ и $k = 0$, после ввода всех 32-отсчетов получим $S(0)/\Delta t = 8$, откуда $|S(0)| = 10^{-6}$ (при точном значении 10^{-6}).

Для вычисления $|S(j\omega_1)|$ на частоте 250 кГц записываем в регистр 4 значение $k = 1$ и, повторив ввод всех отсчетов, получим $|S(j\omega_1)|/\Delta t = 7,214113$ или $|S(j\omega_1)| = 9,017641 \cdot 10^{-7}$ (точное значение $9,003162 \cdot 10^{-7}$).

Аналогично для $f = 500 \text{ кГц}$ ($k = 2$) находим $|S(j\omega_2)|/\Delta t = 5,125831$; $|S(j\omega_2)| = 6,407288 \cdot 10^{-7}$ (точное значение $6,366197 \cdot 10^{-7}$), для $f = 750 \text{ кГц}$ ($k = 3$) $|S(j\omega_3)|/\Delta t = 2,435908$; $|S(j\omega_3)| = 3,044885 \cdot 10^{-7}$ (точное значение $3,044885 \cdot 10^{-7}$) и т. д.

Наличие во входном языке микрокалькулятора типа «Электроника Б3-34» операторов косвенного обращения и адресации позволяет составить программу одновременного вычисления до 11 коэффициентов разложения заданной дискретной функции в тригонометрический ряд Фурье (30). Примером может служить программа 166а/34 ($2\pi/n = PC$, $0 = PD$, $x(i) = RX$).

Программа 166а/34. Разложение в ряд Фурье дискретной функции

↑	1	2	ЗП0	2	ПП	18	sin	ПП	28
1	ПП	18	cos	ПП	28	БП	04	—	$x \neq 0$
50	2	÷	ИПС	ИПД	×	×	В/О	×	КИП0
+	ИП0	1	+	ЗП0	→	КЗП0	→	ИП0	В/О
→	ИП1	+	ЗП1	ИПД	1	+	ЗПД	С/П	БП

При работе с программой 166а/34 отсчеты заданной функции $x(i)$, начиная с $x(0)$ и до $x(n - 1)$, последовательно вводят в регистр X и нажимают клавишу С/П (после ввода $x(0)$ — клавиши В/О и С/П). Обработка каждого очередного отсчета заканчивается выво-

дом на индикатор номера отсчета, который должен быть введен следующим. Результаты выполнения программы находятся: $P1 = na_0/2$, $P2 = na_1/2$, $P3 = nb_1/2$, $P4 = na_2/2$, $P5 = nb_2/2$, $P6 = na_3/2$, $P7 = nb_3/2$, $P8 = na_4/2$, $P9 = nb_4/2$, $PA = na_5/2$, $PB = nb_5/2$.

Вычисление спектров функций, отличных от нуля лишь на некотором интервале периода (или квазипериода), можно ускорить, если после ввода первого отсчета и первого выполнения программы в регистр 3 вместо числа n записать число ненулевых отсчетов заданной функции. Так, в рассмотренном примере перед началом вычисления $S(j\omega)$ на частоте ω_k в регистр 3 вначале записывают 32, а после автоматической обработки первого отсчета записывают 8. В этом случае для получения результата достаточно ввести лишь восемь единичных отсчетов. При повторном использовании программы для другого значения ω_k необходимо перед вводом первого отсчета восстановить число n в регистре 3.

Процесс определения коэффициентов ряда Фурье можно ускорить при параллельном вычислении нескольких пар коэффициентов в соответствии с алгоритмом Герцеля [27]. Согласно этому алгоритму, для определения коэффициентов ряда предварительно определяют значения функций $\cos(2\pi k/n)$ и $\sin(2\pi k/n)$, затем по отсчетам заданной функции y_n, y_{n-1}, \dots, y_1 последовательно вычисляют вспомогательные функции

$$u_j = 2u_{j-1} \cos 2k\pi/n - u_{j-2} + y(n-j); \quad u_0 = 0; \quad u_1 = y_{n-1}$$

и искомые коэффициенты

$$\frac{na_k}{2} = u_{n-1} \cos \frac{2k\pi}{n} - u_{n-2} + y_0; \quad \frac{nb_k}{2} = u_{n-1} \sin \frac{2k\pi}{n}.$$

Таким образом, при предварительном определении входящих в формулы тригонометрических функций вычисления сводятся к операциям сложения и умножения, что при программировании микрокалькуляторов позволяет эффективно использовать стековые регистры.

Необходимость хранения в памяти для каждого вычисления двух значений тригонометрических функций и двух предыдущих значений функций u_i для микрокалькуляторов с 13 регистрами памяти ограничивает число одновременно вычисляемых коэффициентов тремя парами с номерами k_1, k_2 и k_3 .

Вычисления по этому алгоритму реализуются программами 167 и 168, первая из которых используется для подготовки данных.

Программа 167. Подготовка данных для параллельного вычисления коэффициентов тригонометрического ряда

	↑	π	XY	÷	P2	↑	F3	×	↑	sin	P3	XY
cos	2	×	P6	F4		↑	F2	×	↑	sin	P4	XY
cos	2	×	P7	F5		↑	F2	×	↑	sin	P5	XY
cos	2	×	P8	0	P2	←	0	↑	↑	0	←	0
	↔	0	↔	0	↔	0	↔	C/P				

Программа 168. Параллельное вычисление коэффициентов тригонометрического ряда

ПП	7	F6	ПП	P6	F7	ПП	P6	F8	ПП	P6	C/П
ПП	7	F3	×	P3	F6	ПП	5	F4	×	P4	F7
ПП	5	F5	×	P5	F8	ПП	5	C/П	×	2	1/x
×	→	XY	←	+	→	→	↑	B/O	↑	F2	+
P2	←	—	←	—	←	←	—	—	←	↑	B/O

После занесения в УЗП программы 167 в регистры 3, 4 и 5 соответственно вводят номера k_1 , k_2 , k_3 вычисляемых коэффициентов ряда, а в регистр X — число $n/2$. После выполнения программы ($P3 = \sin 2\pi k_1/n$; $P4 = \sin 2\pi k_2/n$; $P5 = \sin 2\pi k_3/n$; $P6 = -2 \cos 2\pi k_1/n$; $P7 = 2 \cos 2\pi k_2/n$; $P8 = 2 \cos 2\pi k_3/n$) в УЗП вводят программу 168, в регистр X — отсчет y_n и нажимают клавиши В/О и С/П. После выполнения программы (время счета около 10 с) в регистр X вводят отсчеты функции (в обратном порядке) y_{n-1}, \dots, y_1 с нажатием клавишей В/О и С/П. Затем вводят y_0 и нажимают клавишу С/П. После выполнения программы $P2 = na_0$; $P3 = nb_{k_1}/2$; $P4 = nb_{k_2}/2$; $P5 = nb_{k_3}/2$; $C2 = na_{k_1}/2$; $C4 = na_{k_2}/2$; $C5 = na_{k_3}/2$ при сохранении содержимого регистров 6 7 и 8.

Программу 168 можно использовать как самостоятельную при вычислении исходных данных в обычном режиме. Для приближенного вычисления спектральной плотности методики использования программ 166 и 168 аналогичны.

Пример 67. Определить модуль спектральной плотности заданной в предыдущем примере функции на частотах 0; 250; 500 и 750 кГц.

Выбрав интервал дискретизации $\Delta t = 0,125$ мкс и определив значения $n = 32$ и $k_1 = 1$, $k_2 = 2$, $k_3 = 3$ для заданных частот, вводим в УЗП программу 167 и заносим $1 = P3$, $2 = P4$, $3 = P5$, $16 = RX$. После выполнения этой программы (время счета около 15 с) вводим в УЗП программу 168 (регистры стека и второй регистр ЗУПВ после выполнения программы 168 оказываются очищены, а в остальных регистрах хранятся числа $0,195090 = P3$, $0,382683 = P4$; $0,555670 = P5$; $1,961570 = P6$; $1,847759 = P7$; $1,662939 = P8$). Затем последовательно заносим в регистр индикации отсчеты функции y_n, \dots, y_1 , нажимая после каждого ввода клавиши В/О и С/П. Вводя последний отсчет и нажимая только клавишу С/П, заканчиваем программу, в результате чего получаем $P2 = S(0)/\Delta t = 8$; $P3 = \text{Im}S(\omega_1)/\Delta t = 4,576588$; $P4 = \text{Im}S(\omega_2)/\Delta t = 5,027341$; $P5 = \text{Im}S(\omega_3)/\Delta t = 2,148277$; $C6 = \text{Re}S(\omega_1)/\Delta t = 5,576596$; $C4 = \text{Re}S(\omega_2)/\Delta t = 1,000001$; $C2 = \text{Re}S(\omega_3)/\Delta t = -1,148279$. По найденным значениям вычисляем $S(j\omega_1) = 9,017654 \times 10^{-7}$; $|S(j\omega_2)| = 6,407291 \cdot 10^{-7}$; $S(j\omega_3) = 3,044883 \cdot 10^{-7}$, что практически совпадает с результатами, полученными в предыдущем примере.

Если тригонометрический ряд используют в качестве интерполирующего, то необходимо вычислять не только коэффициенты ряда, но и значения функций между узлами выбранной табличной модели. Для функций, заданных на интервале $(0; 3)$ четырьмя отсчетами y_0, y_1, y_2, y_3 и с равномерным шагом (предполагается, что функция

периодическая и $y_4 = y_0$), эта задача решается с помощью программы 169, вычисляющей коэффициенты интерполирующего ряда

$$\begin{aligned} y(x) &= \frac{a_0}{2} + a_1 \cos \frac{\pi x}{2} + a_2 \cos \pi x + b_1 \sin \frac{\pi x}{2} = \\ &= \frac{1}{4} \left[y_0 \left(1 + \cos \frac{\pi x}{2} \right) \cos \frac{\pi x}{2} + y_1 \left(1 + \sin \frac{\pi x}{2} \right) \sin \frac{\pi x}{2} - \right. \\ &\quad \left. - y_2 \left(1 - \cos \frac{\pi x}{2} \right) \cos \frac{\pi x}{2} - y_3 \left(1 - \sin \frac{\pi x}{2} \right) \sin \frac{\pi x}{2} \right] \end{aligned}$$

по формулам

$$\begin{aligned} a_0 &= \frac{y_0 + y_1 + y_2 + y_3}{2}; \quad a_1 = \frac{y_0 - y_2}{2}; \quad a_2 = \frac{y_0 + y_2 - y_1 - y_3}{2}; \\ b_1 &= \frac{y_1 - y_3}{2}. \end{aligned}$$

Программа 169. Вычисление суммы членов интерполирующего ряда

\uparrow	π	\times	2	\div	P8	cos	1	$-$	\times	\uparrow	F4
\times	P7	F8	sin	1	—	\times	\uparrow	F5	\times	\uparrow	F7
+	P7	F8	cos	1	+	\times	\uparrow	F2	\times	\uparrow	F7
+	P7	F8	sin	1	+	\times	\uparrow	F3	\times	\uparrow	F7
+	2	\div	C/P	BП	P0						

Перед пуском программы 169 отсчеты функции y_0, y_1, y_2 и y_3 вводят соответственно в регистры 2—5, а в регистр X — значение аргумента, для которого вычисляют функцию $PX = y(x)$.

При определении спектров кусочно-гладких функций по формулам (127) и (130a) методом кусочно-линейной аппроксимации с постоянным шагом коэффициенты η_i можно вычислять по программе 170. В незаполненную часть этой программы записывают операторы вычисления анализируемой (четной или нечетной) функции, заданной на интервале $[-1, 1]$ нормированной переменной v по значению переменной x , хранящейся в регистре 8.

Программа 170. Вычисление коэффициентов спектральных разложений

ПП	FBП	P2	P3	F7	\uparrow	F8	+	P8	ПП	FBП	\uparrow
F2	+	\rightarrow	F3	P2	X Y	P3	X Y	2	\times	\uparrow	\leftarrow
—	2	\times	\uparrow	F7	\div	C/P	F8	1	—	/—/	$x \geqslant 0$
XY	F3	БП	F	\uparrow	B/O

После занесения в УЗП программы 170 в регистр 7 вводят выбранное значение шага h , а регистр 8 очищают. При вычислении коэффициентов η_i четных функций первый вычисленный коэффициент соответствует углу $x_0 = 0$, а последующие — значениям $x_i = ih$. Если вычисляют коэффициенты η_i нечетных функций, то первый коэффициент можно не регистрировать, так как согласно выражениям (127a) и (130a) в дальнейшем этот коэффициент умножается на значение функции $\sin x_0 v \equiv 0$.

В программе 170 предусмотрено автоматическое выполнение условия $y_{i+1} = y_i$ при $x_i > 1$, позволяющее определить значение коэффициентов η_i на границах интервала определения функции. Первое обращение к программе реализуется нажатием клавиши В/О и С/П, последующие — только клавиши С/П. После каждого выполнения программы на индикатор выводится удвоенное значение коэффициента η_i с обратным знаком, т. е. $-2\eta_i$, что облегчает дальнейшие вычисления по формулам (127а) или (130а).

Например, для вычисления коэффициентов η_i при заданной в примере 41 функции дополняем программу 170 фрагментом вычисления четной части $\dots |F8| x^2 |1| - | / - | / B/O |$ и при шаге $h = 0,25$ получаем $-2\eta_0 = 0,5$; $-2\eta_1 = 1$; $-2\eta_2 = 1$; $-2\eta_3 = 1$; $-2\eta_4 = -3,5$ и нули при дальнейших пусках. Для нечетной части вводим в программу фрагмент $\dots |F8| \uparrow | \pi | \times | \sin | / - | / B/O |$ и при шаге $h = 0,2$ получаем (пропустив $-2\eta_0$) $-2\eta_1 = -2,245141$; $-2\eta_2 = -3,633712$; $-2\eta_3 = -3,632711$; $-2\eta_4 = -2,24514$; $-2\eta_5 = 5,877848$ (время вычисления каждого значения около 8 с).

Вычисление коэффициентов α_q и β_q при параболической аппроксимации по формулам (128) даже для сравнительно простых функций на микрокалькуляторе типа «Электроника Б3-21» в программируемом режиме практически невозможно, а решение задачи по частям громоздко.

При автоматическом вычислении суммы членов тригонометрического ряда или спектральной плотности по формулам (127а), (128а) или (130) ограничения на число суммируемых членов ряда в основном связаны с емкостью ЗУПВ, так как использование стека (в связи с возможностью неконтролируемого переполнения) затруднено. Однако, если суммируемый ряд содержит до семи членов (включая $a_0/2$) и, следовательно, максимальное значение аргумента функций $\sin x$ и $\cos x$ не превышает 6π , то, согласно данным табл. 40, можно использовать, по крайней мере, регистр С1 стека.

Например, для вычисления суммы членов ряда

$$y(x) = A_0 + \sum_{k=1}^3 (A_k \cos 2\pi kx + B_k \sin 2\pi kx)$$

можно использовать программу 171 с исходными данными $A_1 = P2$, $A_2 = P3$, $A_3 = P4$, $B_1 = P5$, $B_2 = P6$, $B_3 = P7$ (значение A_0 записывают в свободной части программы перед операторами + С/П).

Программа 171. Вычисление суммы членов ряда Фурье с $k=3$

\uparrow	π	\times	6	\times	P8	e^{ix}	F7	\times	\rightarrow	F4	ПП
8	F8	3	\div	P8	e^{ix}	F5	ПП	P9	F2	ПП	8
F8	2	\times	P8	e^{ix}	F6	ПП	P9	F3	ПП	8	\leftarrow
...	+	C/П	БП	P0
...	\uparrow	F8	\cos	\times	\uparrow	\leftarrow	\rightarrow	+ \rightarrow	B/O

В программе 171 регистр 8 использован в качестве вспомогательного, а в регистре стека С1 накапливается сумма вычисленных членов. При изменении размещения заключительной части подпрограммы в УЗП по сравнению с приведенным необходимо соответственно изменить адреса обращений к подпрограмме в остальной части программы.

Перед пуском программы 171 в регистр X вводится значение нормированной (делением на T) переменной x , для которого необходимо вычислить сумму членов ряда.

Свободные ячейки УЗП можно использовать для формирования оператора присвоения переменной x нового значения, обеспечив упрощение повторных вычислений при постоянном значении шага Δx , также записываемого непосредственно в текст программы. Для этого содержимое регистра 8 делится на 4 и к нему добавляется выбранное значение Δx . Например, при $\Delta x = 0,1$ программу 171 дополняют фрагментом С/П F8 4 $\div \uparrow \pi \div 0,1 +$ БП Р0, начинающимся с шага по адресу 61.

Для вычисления значений тригонометрического многочлена вида

$$y(x) = C_0 \sum_{k=1}^3 C_k \cos(2\pi k x + \varphi_k)$$

следует использовать программу 172 с исходными данными $C_1 = P2$, $\varphi_1 = P3$, $C_2 = P4$, $\varphi_2 = P5$, $C_3 = P6$, $\varphi_3 = P7$. Значение коэффициента $C_0 \neq 0$ записывается непосредственно в заключительной части программы. Значение аргумента, при котором вычисляют сумму ряда, вводится в регистр X. При необходимости в заключительной части программы можно организовать автоматическое изменение аргумента на величину Δx описанным выше способом.

Программа 172. Вычисление суммы членов косинусоидального ряда

\uparrow	π	\times	6	\times	P8	\uparrow	F7	$+$	cos	\uparrow	F6
\times	\rightarrow	F8	3	\div	P8	\uparrow	F3	$+$	cos	\uparrow	F2
\times	\uparrow	\leftarrow	$+$	\rightarrow	F8	2	\times	\uparrow	F5	$+$	cos
\uparrow	F4	\times	\uparrow	\leftarrow	$+$	\dots	\dots	\dots	C/П	BП	P0

Число независимых коэффициентов можно сократить нормированием членов ряда по одному из его коэффициентов и смещением начала отсчета аргумента на величину $x_0 = \varphi_1$. Например, для рассматриваемого ряда подобным образом получим выражение

$$\frac{y(x - x_0)}{C_1} = D_0 + \cos 2\pi x + D_2 \cos(4\pi x + \psi_2) + D_3 \cos(6\pi x + \varphi_3) + \\ + D_4 \cos(8\pi x + \psi_4),$$

где $D_k = C_k / C_1$; $\psi_k = \varphi_k - k\varphi_1$. В этом случае, записывая значение D_0 непосредственно в текст, нетрудно составить программу 173 для суммирования ряда с пятью членами (включая D_0) при

исходных данных $D_2 = P_2$, $\Psi_2 = P_3$, $D_3 = P_4$, $\Psi_3 = P_5$, $D_4 = P_6$, $\Psi_4 = P_7$. В остальном использование программ 172 и 173 аналогично.

Программа 173. Вычисление суммы членов нормированного ряда Фурье

\uparrow	π	\times	8	\times	P8	\uparrow	F7	$+$	cos	\uparrow	F6
\times	\rightarrow	F8	2	\div	\uparrow	F3	$+$	cos	\uparrow	F2	\times
\uparrow	\leftarrow	$+$	\rightarrow	F8	4	\div	P8	cos	\uparrow	\leftarrow	$+$
\rightarrow	F8	3	\times	\uparrow	F5	$+$	cos	\uparrow	F4	\times	\uparrow
\leftarrow	$+$	\dots	\dots	\dots	\dots	\dots	\dots	\dots	C/P	B/P	P0

При равенстве нулю одного из членов ряда для $k \leq 4$ можно включить в программу 173 значение еще одного члена ряда, для чего следует предусмотреть соответствующее изменение множителя k у этого дополнительного члена.

Пример 68. Составить программу вычисления многочлена

$$y(x) = C_0 + C_1 \cos(2\pi x + \varphi_1) + C_2 \cos(4\pi x + \varphi_2) + \\ + C_3 \cos(6\pi x + \varphi_3) + C_5 \cos(10\pi x + \varphi_5)$$

при различных значениях коэффициентов C для определения формы аппроксимирующей функции.

Так как интерес представляет не абсолютное значение многочлена, а форма результирующей кривой, то опустим постоянную составляющую C_0 , нормируем многочлен по коэффициенту C_1 и сместим начало отсчета на величину $x_0 = \varphi_1$. Тогда получим

$$z(x - x_0) = \cos 2\pi x + D_2 \cos(4\pi x + \varphi_2) + \\ + D_3 \cos(6\pi x + \varphi_3) + D_5 \cos(10\pi x + \varphi_5).$$

Разместив в памяти коэффициенты $D_2 = P_2$, $\varphi_2 = P_3$, $D_3 = P_4$, $\varphi_3 = P_5$, $D_5 = P_6$, $\varphi_5 = P_7$ и введя в основную программу 173 подпрограмму ... \times \uparrow \leftarrow \rightarrow F3 B/O, получим программу 174. При необходимости в свободной части этой программы можно организовать денормировку результата и учесть коэффициент A_0 .

Программа 174. Определение формы аппроксимирующей кривой

\uparrow	π	\times	2	\times	P8	cos	\rightarrow	F8	2	\times	\uparrow
F3	$+$	cos	\uparrow	F2	P\P	B\P	3	\times	\uparrow	F5	$+$
cos	\uparrow	F4	P\P	B\P	5	\times	\uparrow	F7	$+$	cos	\uparrow
F6	P\P	B\P	\leftarrow	C/P	\times	\uparrow	\leftarrow	$+$	\rightarrow	F8	B/O

В инженерных приложениях (в частности, в радиотехнике) встречается задача определения спектрального состава реакции на гармоническое воздействие в нелинейной безынерционной системе, характеристика $y(x)$ которой задана графически или таблично. В случае слабой нелинейности при инженерных расчетах часто используют метод пяти ординат с вычислением постоянной состав-

ляющей и первых четырех гармоник по пяти отсчетам функции (рис. 42, а) согласно формулам:

$$Y_0 = ((y_{\max} + y_{\min}) + 2(y_1 + y_2))/6;$$

$$Y_1 = ((y_{\max} - y_{\min}) + (y_1 - y_2))/3;$$

$$Y_2 = ((y_{\max} + y_{\min}) - 2y_0)/4;$$

$$Y_3 = ((y_{\max} - y_{\min}) - 2(y_1 - y_2))/6;$$

$$Y_4 = ((y_{\max} + y_{\min}) - 4(y_1 + y_2) + 6y_0)/12.$$

Вычисления по приведенным формулам часто сочетают с оценкой коэффициента нелинейных искажений

$$k_{\text{нел}} = \left(\sqrt{Y_2^2 + Y_3^2 + Y_4^2} \right) / Y_1.$$

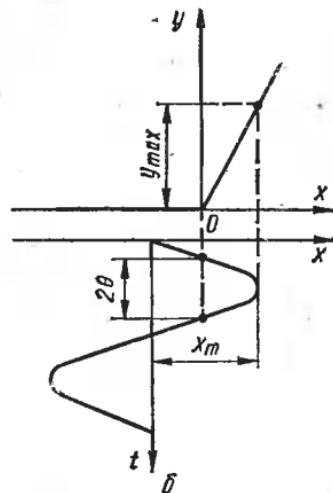
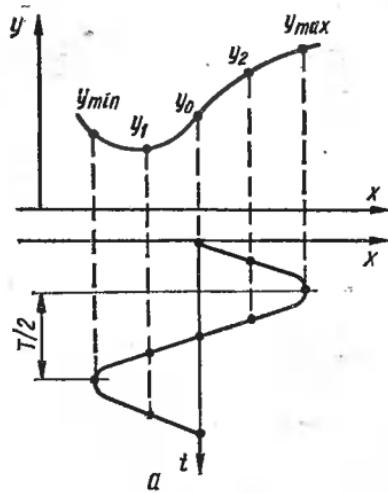


Рис. 42. Графические модели характеристик нелинейных систем

Для неоднократного вычисления по этим формулам целесообразно использовать программу 175 ($y_0 = P6$, $y_2 = P7$, $y_1 = P8$, $y_{\max} = PY$, $y_{\min} = PX$). После выполнения программы (время счета около 10с) результаты вычислений хранятся в регистрах $P2 = Y_0$, $P3 = Y_1$, $P4 = Y_2$, $P5 = Y_3$, $P6 = Y_4$, $PX = k_{\text{нел}}$.

Программа 175. Вычисление амплитуд гармоник и коэффициента нелинейных искажений методом пяти отсчетов

—	P5	—	+	2	÷	↑	F6	—	→	F7	+
↑	F8	+	3	÷	P2	←	2	÷	P4	↑	F2
—	↑	F6	+	P6	F8	↑	F7	—	P7	↑	F5
+	3	÷	P3	↑	F7	—	2	÷	P5	x^2	↑
F6	x^2	+	↑	F4	x^2	+	V	↑	F3	÷	C/П

По программе 175 для $y_{\min} = 0,5$; $y_2 = 4,5$; $y_0 = 9$; $y_1 = 10$; $y_{\max} = 8$ получаем $k_{\text{нел}} = 0,5709627$; $Y_0 = 6,25$; $Y_1 = 4,333333$; $Y_2 = -2,375$; $Y_3 = -0,5833333$; $Y_4 = 0,375$.

Если характеристика нелинейной безынерционной системы аппроксимирована отрезками прямых (рис. 42, б), то при гармоническом воздействии спектральный состав реакции определяют с помощью коэффициентов Берга

$$\alpha_0(\theta) = \frac{\sin \theta - \theta \cos \theta}{\pi(1 - \cos \theta)};$$

$$\alpha_k(\theta) = \frac{1}{\pi(1 - \cos \theta)} \left[\frac{\sin(k-1)\theta}{(k-1)k} - \frac{\sin(k+1)\theta}{(k+1)k} \right],$$

где θ — угол отсечки воздействия.

Амплитуды гармоник определяются по коэффициентам Берга согласно формулы $Y_k = Y_{\max} \alpha_k(\theta)$, где Y_{\max} — максимальное значение реакции. Вычисление коэффициентов Берга для заданных значений номера гармоники k и угла отсечки θ (в градусах) обеспечивается программой 176 ($k = P2$, $\theta = PX$): Результат вычислений (время счета около 15 с) высвечивается на индикаторе.

Программа 176. Вычисление коэффициентов Берга

P3	1	8	0	\div	\uparrow	π	XY	\times	P4	cos	\times
—	P5	F2	$x=0$	\rightarrow	F4	\uparrow	\cos	\times	XY	sin	XY
БП	F7	1	+	ПП	P8	P8	F2	1	—	$x=0$	6
F4	БП	ВП	ПП	P8	\uparrow	F8	—	\uparrow	F5	\div	C/P
P7	\uparrow	F4	\times	e^{ix}	F7	\div	\uparrow	F2	\div	B/O	

Правильность ввода программы 176 можно проверить по вычислению $\alpha_0(30^\circ) = 0,1105983$; $\alpha_1(30^\circ) = 0,2152232$; $\alpha_2(90^\circ) = 0,2122065$.

Иногда удобно непосредственно вычислять амплитуды гармоник реакции на гармоническое воздействие с амплитудой x_m при крутизне S участка характеристики с ненулевым значением функции (рис. 42, б) или коэффициенты $\gamma_k(\theta) = (1 - \cos \theta) \alpha_k(\theta)$. Для вычисления $Y_k = f(X_m)$ можно использовать программу 177 ($k = PX$, $\theta = P3$, $S = P4$, $X_m = P5$).

Программа 177. Вычисление амплитуд гармоник при кусочно-линейном преобразовании

P2	$x \neq 0$	FBП	1	—	$x \neq 0$	P/-/	P6	2	+	\uparrow	F3
\times	sin	XY	\div	P7	F6	\uparrow	F3	\times	sin	XY	\div
\uparrow	F7	—	\uparrow	F2	\div	БП	F8	F3	e^{ix}	\times	\uparrow
F3	XY	—	БП	F8	F3	\uparrow	cos	\times	XY	sin	XY
—	\uparrow	F4	\times	\uparrow	F5	\times	\uparrow	π	\div	C/P	

После выполнения программы 177 (время счета около 12 с) все исходные данные сохраняются в своих регистрах, а результат C_k высвечивается на индикаторе. Для вычисления $\gamma_k(\theta)$ достаточно в регистры 5 и 4 записать единицы. Для $\theta = \pi/3$; $S = 3,1415927$; $X_m = 1$ по программе 176 получим $C_0 = 0,3424266$; $C_1 = 0,6141847$; $C_2 = 0,4330126$; $C_3 = 0,2165063$; $C_4 = 0,04330128$.

При воздействии на нелинейную безынерционную систему суммы двух гармонических воздействий с частотами f_c и f_g реакция имеет составляющие с комбинационными частотами $f_k = |nf_c \pm mf_g|$. Иногда необходимо определить частоты $f_c \in [f_{\min}, f_{\max}]$ одного из воздействий, которые совместно с фиксированной частотой f_g второго воздействия образуют комбинационные составляющие реакции определенной частоты f_k . Подобная задача, в частности, возникает при определении побочных каналов приема в супергетеродинных приемниках.

В рассматриваемом случае вычисления основаны на подстановке в формулу $f_c = |mf_g \pm f_k|/n$ различных сочетаний целых чисел n и m , при которых удовлетворяется условие $f_{\min} < f_c < f_{\max}$. Перебор допустимых сочетаний этих чисел можно организовать в соответствии со следующим алгоритмом:

1. Принять $m + n = 1$, $m = 0$, $n = 1$ и перейти к п. 2.
2. Принять $m + n = m + n + 1$, $m = 0$, $n = (m + n) - m$, проверить условие $f_{\min} < nf_c < f_{\max}$ и перейти к п. 3.
3. Принять $m = m + 1$, $n = n - 1$ и перейти к п. 4.
4. Если $n = 0$, то перейти к п. 2, иначе к п. 3.

Подобный алгоритм перебора по возрастанию порядка $m + n$ комбинационных частот использован в программе 178 ($f_g = P2$, $f_k = P3$, $f_{\min} = P4$, $f_{\max} = P5$, $0 = P8$). После выполнения программы на индикаторе высвечивается очередное значение частоты f_c , а в регистрах 6 и 7 хранятся соответствующие числа m и n . Выполнение программы повторяют (первый пуск — нажатием клавиш В/О и С/П, последующих — только клавиши С/П) до вычисления всех представляющих интерес частот. Программа прекращает перебор чисел при $(mf_g - f_k)/n > f_{\max}$.

Программа 178. Вычисление частоты воздействия по заданным комбинационным частотам реакции

F8	1	—	P8	0	P6	↑	F8	+	/—/	P7	$x \neq 0$
P0	F2	×	↑	↔	F3	—	$x \geq 0$,	ΠΠ	P7	F5
—	$x < 0$	P0	БП	P/—/	/—/	ΠΠ	P7	→	↑	F3	+
ΠΠ	P7	1	+	БП	↑	↑	F7	÷	↑	F4	—
$x \geq 0$	sin	F5	—	$x < 0$	P+	XY	C/Π	F6	B/O		

Например, для вычисления m и n , а также частот f_c в интервале от 2985 до 3085 кГц, образующих с $f_g = 3500$ кГц комбинационную частоту $f_k = 465$ кГц, после занесения программы 178 в УЗП следует выполнить операции, описываемые следующим предложением: 0 P8 3500 P2 465 P3 2985 P4 3085 P5 B/O C/Π (3035) F6 (1) F7 (1) C/Π (2994, 166) F6 (5) F7 (6) C/Π (3066, 428) F6 (6) F7 (7) C/Π (3004, 375) F6 (7) F7 (8)...

Рассмотренный алгоритм перебора можно использовать и при другой постановке задачи с комбинационными частотами, а также в ряде других комбинаторных задач.

7. АВТОМАТИЗАЦИЯ СТАТИСТИЧЕСКИХ РАСЧЕТОВ

Составление программ для вычисления вероятностей по формулам табл. 4 при наиболее часто встречающихся на практике законах распределения не вызывает существенных затруднений. Исключение составляют лишь некоторые из них, например, закон Стьюдента или гамма-распределение. Покажем, как можно составить программы автоматических вычислений в этих случаях.

При распределении по закону Стьюдента в формуле

$$p(x) = \frac{1}{V\pi N} \frac{\Gamma\left(\frac{N+1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \left(1 + \frac{x^2}{N}\right)^{-\frac{N+1}{2}},$$

где $N \geq 2$ заменим

$$\Gamma\left(\frac{N+1}{2}\right) = 2\Gamma\left(\frac{N+3}{2}\right)/(N+1); \quad \Gamma\left(\frac{N}{2}\right) = 2\Gamma\left(\frac{N+2}{2}\right)/N$$

и по формуле Стирлинга найдем преобразование гамма-функции

$$\Gamma\left(\frac{N+3}{2}\right) = \sqrt{4\pi/(N+3)} \left(\frac{N+3}{2e}\right)^{(N+3)/2} H\left(\frac{N+3}{2}\right);$$

$$\Gamma\left(\frac{N+2}{2}\right) = \sqrt{4\pi/(N+2)} \left(\frac{N+2}{2e}\right)^{(N+2)/2} H\left(\frac{N+2}{2}\right),$$

где для вычисления многочленов H ограничимся двумя членами $H(z) = 1 - 1/12z$. Это позволяет представить исходную функцию приближенным выражением

$$p(x) \approx \sqrt{\frac{N(N+2)}{17,08}} \left(\frac{N+3}{N+2}\right)^N \frac{(6N+19)}{(N+1)(6N+13)} \left(1 + \frac{x^2}{N}\right)^{-\frac{N+1}{2}},$$

которое реализуется программой 179 ($x = P2$, $N = P3$, $PX = p(x)$).

Программа 179. Вычисление плотности вероятности при распределении Стьюдента

F3	3	+	1	-	P4	÷	↑	F3	XY	x^y	×
↑	F4	×	1	7	,	0	8	÷	V	P5	F3
1	+	P6	6	×	7	+	6	+	÷	↑	F6
×	↑	F5	÷	P5	F2	x^2	↑	F3	÷	1	+
↑	F6	XY	x^y	V	↑	F5	×	1/x	C/P	БП	P7

При повторных пусках программы 179 в регистр X вводят значение x и нажимают только клавишу C/P. В наихудшем случае результат вычислений по программе (время счета около 15 с) имеет не менее четырех верных знаков. При $N = 2$, $x = 0$ по программе 179 получим $p(x) = 0,3535989$.

Для гамма-распределения

$$p(x) = \begin{cases} 0 & \text{при } x \leq 0; \\ \frac{x^\alpha e^{-x/\beta}}{\beta^{\alpha+1} \Gamma(\alpha+1)} & \text{при } x > 0, \alpha > -1, \beta > 0 \end{cases}$$

можно принять

$$\Gamma(\alpha+1) = \frac{\Gamma(\alpha+3)}{(\alpha+1)(\alpha+2)} \approx \frac{1}{(\alpha+1)(\alpha+2)} \sqrt{\frac{2\pi}{\alpha+3}} \left(\frac{\alpha+3}{e}\right)^{\alpha+3} \times \\ \times H(\alpha+3).$$

Ограничивааясь двумя членами укороченного многочлена $H(\alpha+3) = 1 + 1/12(\alpha+3)$, приводим исходную формулу при $x > 0$ к виду

$$p(x) \approx \frac{(\alpha+1)(\alpha+2)}{\beta((\alpha+3)/e)^{\alpha+3} [2 + 1/6(\alpha+3)]} \sqrt{\frac{2(\alpha+3)}{\pi}} \left(\frac{x}{\beta}\right)^\alpha e^{-x/\beta}.$$

Полученная приближенная формула реализуется программой 180 ($x = P2$, $\alpha = P3$, $\beta = P4$).

Программа 180. Вычисление плотности вероятности при гамма-распределении

F3	1	+	1	+	\times	P5	F3	3	+	P6	1
e ^x	\div	x^α	P7	π	\div	2	\times	$\sqrt[3]{}$	\uparrow	F5	\times
\uparrow	F7	\div	\uparrow	F4	\div	P5	F6	6	\times	1/x	2
\uparrow	\uparrow	F5	\div	P5	F2	\uparrow	F4	\div	P2	e^x	P6
F3	XY	x^α	\uparrow	F5	\div	\uparrow	F6	\div	C/P	BП	BП

Результат вычислений по программе 180 выводится на индикатор не менее, чем с четырьмя верными знаками, причем исходные данные сохраняются в «своих» регистрах. При повторном пуске программы достаточно ввести в регистр X значение x и нажать клавишу С/П. Время счета в этом случае не превышает 8—10с. Для $\alpha = \beta = 1$ по программе 180 получим $p(0,4) = 0,2681741$; $p(1) = 0,3679425$.

Характеристики случайного числа вычисляют с помощью программы 181 ($N = P7$, $x_i = PX$, $P2 = \tilde{m}_x$, $P3 = \tilde{\sigma}_x^2$, $P4 = \tilde{\sigma}_x$, $P5 = \tilde{\sigma}_m^2$, $P6 = \sigma_m$). Для пуска программы после ввода первого отсчета нажимают клавиши В/О и С/П, для последующих пусков — только клавишу С/П.

Программа 181. Вычисление характеристик случайного числа

P3	x^2	P4	F7	P2	1	$-$	P6	0	\uparrow	F3	$+$
P3	XY	x^2	\uparrow	F4	$+$	P4	F2	1	\div	P2	$x \neq 0$
F,	C/P	\uparrow	1	F3	\uparrow	F7	\div	P2	x^2	\uparrow	F7
\times	\uparrow	F4	$-$	\uparrow	F6	\div	/—/	P3	\uparrow	F7	\div
P5	$\sqrt[3]{}$	P6	F3	$\sqrt[3]{}$	P4	C/P					

После выполнения программы 181 на индикатор выводится величина $\sigma_x = P_4$, но для вывода на индикатор любой другой вычисляемой величины достаточно между операторами P_4 и C/P программы записать соответствующий оператор вызова FN . В процессе вычислений обработка каждого отсчета заканчивается высвечиванием числа оставшихся необработанными отсчетов и результаты вычислений по программе заносятся в соответствующие регистры после обработки последнего отсчета.

Для одновременного вычисления первых четырех моментов случайного числа предназначена программа 182 ($0 = P_8$, $x_i = P_X$).

Программа 182. Вычисление четырех начальных моментов

P2	F8	$x \neq 0$	P3	$1/x$	\uparrow	\leftarrow	\times	C/P	\leftarrow	\times	C/P
\leftarrow	\times	C/P	\leftarrow	\times	C/P	\leftarrow	0	\leftarrow	0	\leftarrow	0!
\leftarrow	\leftarrow	\leftarrow	F2	\uparrow	\times	P2	\times	P3	\times	P4	\leftarrow
\leftarrow	\leftarrow	\leftarrow	F2	\uparrow	\leftarrow	\uparrow	F3	\leftarrow	\uparrow	F4	
\leftarrow	\leftarrow	\leftarrow	F8	1	\leftarrow	\uparrow	P8	C/P	BП	F,	

При работе с программой 182 очередные значения x_i вводят в регистр X нажатием клавиш B/O и C/P после первого ввода и только клавиши C/P после остальных вводов. Каждое выполнение программы оканчивается высвечиванием числа обработанных отсчетов. После выполнения программы для последнего отсчета нажимают клавиши B/O и C/P и регистрируется высвечиваемое значение момента m_1 . Дополнительные нажатия клавиши C/P приводят к высвечиванию значений m_2 , m_3 и m_4 . Например, после обработки всех 26 отсчетов случайного числа, приведенных в табл. 10, получим $m_1 = 2,192307$; $m_2 = 6,229338$; $m_3 = 19,86638$; $m_4 = 67,62200$.

При необходимости вычисления доверительного интервала следует использовать программу 183, позволяющую одновременно определить следующие оценки характеристик обрабатываемого ансамбля: среднее $m_x = P_2$, среднеквадратичное отклонение среднего $\tilde{\sigma}_m = P_3$, полуширину доверительного интервала $\Delta\tilde{m} = t\tilde{\sigma}_m = P_4$, верхнюю $\tilde{m}_{\max} = \tilde{m}_x + \Delta\tilde{m} = P_5$ и нижнюю $\tilde{m}_{\min} = \tilde{m}_x - \Delta\tilde{m} = P_6$ границы доверительного интервала.

Программа 183. Вычисление характеристик обрабатываемого ансамбля

P3	x^2	P4	F7	P2	1	—	P6	0	\uparrow	F3	+
P3	XY	x^2	\uparrow	F4	$+$	P4	F2	1	$\frac{—}{}$	P2	$x \neq 0$
F,	C/P	БП	1	F3	\uparrow	F7	\div	P2	F4	\uparrow	F7
\div	\uparrow	F2	x^2	$\frac{—}{}$	\uparrow	F6	\div	V	P3	\uparrow	F8
\times	P4	\uparrow	F2	XY	$+$	P5	F4	$\frac{—}{}$	P6	C/P	

Перед пуском программы 183 очищают регистр 3, число N обрабатываемых данных и коэффициент t (определенный по табл. 1) вводят соответственно в регистры 7 и 8, а обрабатываемые числа вводятся

поочередно в регистр X с нажатием клавиши С/П (клавишей В/О и С/П после первого ввода). Последнее выполнение программы приводит к занесению результатов вычислений в указанные выше регистры.

При вычислении доверительного интервала по программе 184 не требуется предварительной записи числа N обрабатываемых отсчетов, так как оно автоматически подсчитывается при их вводе в регистр X после нажатия клавиши С/П (В/О и С/П после первого ввода). После ввода последнего отсчета и выполнения программы дополнительно нажимают клавиши В/О и С/П, что приводит к занесению результатов вычислений в регистры: $P2 = \tilde{m}_x$, $P3 = \tilde{\sigma}_x^2$, $P6 = \tilde{m}_x - \Delta\tilde{m}$, $P7 = \tilde{m}_x + \Delta\tilde{m}$.

Программа 184. Вычисление доверительного интервала

\leftarrow	F3	$x \neq 0$	F6	\uparrow	F7	\div	P2	x^2	\uparrow	F7	\times
\uparrow	F4	—	\uparrow	F6	\div	/—/	P3	\uparrow	F7	\div	V
\uparrow	F8	\times	P5	F2	\uparrow	F5	+	P7	F5	—	P6
C/П	0	P4	P7	\rightarrow	\uparrow	F3	+	P3	XY	x^2	\uparrow
F4	+	P4	F7	P6	1	\uparrow	P7	C/П	БП	ВП	

Корреляция между двумя случайными величинами y_i и x_i оценивается в соответствии с формулой (13e) по программе 185. При использовании этой программы очищают регистр 5, в регистр 8 заносят число N пар чисел y_i и x_i , подлежащих обработке, стек очищают вводом нулей и пары значений y_i и x_i вводят соответственно в регистры Y и X (после ввода числа y_i в регистр X нажимают клавишу оператора \uparrow для засылки операнда в регистр Y) с нажатием клавищей В/О и С/П после ввода каждой пары чисел. После ввода последней пары и ее обработки нажимают только клавишу С/П, что приводит к высвечиванию значения коэффициента корреляции с обратным знаком.

Программа 185. Вычисление коэффициента корреляции

P2	\times	ПП	ВП	F3	P4	F2	\uparrow	ПП	ВП	F5	\uparrow
\leftarrow	+	P5	C/П	ПП	8	P6	\leftarrow	ПП	8	\uparrow	F6
\times	$\sqrt{ }$	P7	F3	\uparrow	F4	\times	\uparrow	F8	\div	\uparrow	F5
—	\uparrow	F7	\div	C/П	\leftarrow	+	P3	XY	x^2	XY	\leftarrow
+	\leftarrow	B/O	\leftarrow	x^2	\uparrow	F8	\div	\uparrow	\leftarrow	—	B/O

При вводе $(y_i, x_i) = (1; 2), (0,5; 3), (4; 1), (-2; 0), (3; -1)$ после выполнения программы 185 высвечивается $PX = 0,1354570$.

Значение автокорреляционной функции при использовании микрокалькулятора с 13—15 регистрами памяти можно оценить не более чем в шести равноотстоящих точках, включая начальную. По программе 186 для вводимой в регистр X последовательности чисел x_i (с нажатием клавишей В/О и С/П только после первого

ввода) вычисляются следующие величины: $P8 = \sum x_i$, $C1 = \sum x_i^2$,
 $C2 = \sum_{i=1}^{n-1} x_i x_{i-1}$, $C3 = \sum_{i=2}^{n-2} x_i x_{i-2}$, $C4 = \sum_{i=3}^{n-3} x_i x_{i-3}$, $C5 = \sum_{i=4}^{n-4} x_i x_{i-4}$,
 $C6 = \sum_{i=5}^{n-5} x_i x_{i-5}$.

Программа 186. Вычисление автокорреляционной функции

P2	↑	F8	+	P8	F2	ПП	P6	F3	ПП	P6	F4
ПП	P6	F5	ПП	P6	F6	ПП	P6	F7	ПП	P6	F6
P7	F5	P6	F4	P5	F3	P4	F2	P3	C/P	БП	P0
↑	F2	×	↑	←	+	↑	→	XY	←	B/O	

В процессе выполнения программы 186 в регистры ЗУПВ заносятся $P2 = x_i$, $P3 = x_{i-1}$, $P4 = x_{i-2}$, $P5 = x_{i-3}$, $P6 = x_{i-4}$, $P7 = x_{i-5}$. Стек должен быть очищен перед первым пуском программы. Время обработки одного отсчета около 15 с. Для $x_i = 1; 3; -2; 1,5; -0,5; 0; 1; -1,5; 2; -1; 0,5; -1,5$ по программе 186 получим $P8 = 2,5$; $C1 = 27,25$; $C2 = -14,5$; $C3 = 9$; $C4 = -2,5$; $C5 = -3$; $C6 = 8$.

Для вычисления нормированной автокорреляционной функции $\rho(k\Delta t) = R(k\Delta t)/\sigma_x^2$ после вычислений по программе 186 в УЗП заносят программу 187, реализующую вычисления по формуле (13ж), а в регистр 7 вводят число N обрабатываемых отсчетов.

Программа 187. Вычисление нормированной автокорреляционной функции

F8	↑	F7	÷	x^2	P8	ПП	6	ПП	6	ПП	6
ПП	6	ПП	6	ПП	6	←	P2	ПП	F/-/	ПП	F/-/
ПП	F/-/	ПП	F/-/	ПП	F/-/	ПП	F/-/	0	C/P	↑	F2
÷	←	B/O	F7	1	—	P7	÷	P6	←	XY	÷
↑	F8	—	↑	F6	×	↑	→	XY	←	B/O	

После выполнения программы 187 на индикаторе высвечивается цифра 0, нормированные значения автокорреляционной функции хранятся в стеке: $C1 = \tilde{\rho}(0) = 1$, $C2 = \tilde{\rho}(\Delta t)$, $C3 = \tilde{\rho}(2\Delta t)$, $C4 = \tilde{\rho}(3\Delta t)$, $C5 = \tilde{\rho}(4\Delta t)$, $C6 = \tilde{\rho}(5\Delta t)$. Кроме того, вычисляются $P2 = \tilde{\sigma}_x^2$ и квадрат постоянной составляющей $P8 = (\tilde{m}_x)^2$. При выполнении программы 187 по исходным данным, вычисленным в контрольном примере для программы 186, получим $C1 = 1$; $C2 = -0,6163744$; $C3 = 0,391689$; $C4 = -0,1486993$; $C5 = -0,1967858$; $C6 = 0,5278752$; $P2 = 2,429924$; $P8 = 0,04340277$.

При моделировании сложных функциональных зависимостей, не поддающихся аппроксимации простыми аналитическими выражениями, по результатам эксперимента используют операции сглаживания с помощью многочленов, приближающих выбранные группы экспериментальных отсчетов по методу наименьших квадратов.

Сглаживание уменьшает влияние случайных отклонений результатов эксперимента («шумов»).

Программы 188 и 189 предназначены для сглаживания результатов эксперимента соответственно при линейной аппроксимации для пяти точек согласно формуле (111б) и при квадратичной аппроксимации для семи точек по формуле (111в). Иногда сглаживание целесообразно повторить несколько раз.

Программа 188. Сглаживание при линейной аппроксимации по пяти точкам

$$\begin{array}{ccccccccc} P_6 & C_x & C/\Pi & P_5 & C_x & C/\Pi & P_4 & C_x & C/\Pi \\ P_2 & F_6 & \uparrow & F_5 & P_6 & + & \uparrow & F_4 & P_5 \\ P_4 & + & \uparrow & F_2 & P_3 & + & 5 & \div & C/\Pi \\ & & & & & & & & B\Pi \end{array}$$

Программа 189. Сглаживание при квадратичной аппроксимации по семи точкам

$$\begin{array}{cccccccccc} P_8 & F_2 & 2 & \times & \leftarrow & F_3 & P_2 & 3 & \times & \leftarrow & F_4 & P_3 \\ 6 & \times & \leftarrow & F_5 & P_4 & 7 & \times & \leftarrow & F_6 & P_5 & 6 & \times \\ \leftarrow & F_7 & P_6 & 3 & \times & \leftarrow & F_8 & P_7 & 2 & \times & /-/- & \uparrow \\ \rightarrow & + & \uparrow & \rightarrow & + & \uparrow & \rightarrow & + & \uparrow & \rightarrow & + & \uparrow \\ \rightarrow & + & \uparrow & \rightarrow & - & 2 & 1 & \div & C/\Pi & B\Pi & P_0 & \uparrow \end{array}$$

При работе с программой 188 в регистр X последовательно вводят значения отсчетов и нажимают клавишу C/Π (В/О и C/Π после первого ввода). После первых четырех пусков программы высвечивается цифра 0, указывающая на необходимость вычислений по формулам сглаживания для краев табличной модели. Начиная с пятого пуска программы, на индикатор выводится сглаженное значение x_{i-2} , где i — номер отсчета, введенного последним.

Работа с программой 189 отличается отсутствием автоматической индикации указаний на необходимость уточнения значений отсчетов на краях таблицы. Сглаженные значениячитываются с индикатора после седьмого пуска программы, причем высвечиваются отсчеты x_{i-3} . Время обработки одного отсчета по программам 188 и 189 составляет около 6 и 10 с соответственно.

В ряде случаев необходимо вычислить производную исследуемой функциональной зависимости по табличной модели, содержащей случайные погрешности. Для этого предварительно сглаживают экспериментальные данные, а затем используют формулы численного дифференцирования или алгоритмы, обеспечивающие выполнение обеих операций одновременно. Так, при сглаживании многочленом третьей степени по пяти точкам производная определяется выражением

$$y'_i = ((y_{i+1} - y_{i-1}) \cdot 2 : 3 - (y_{i+2} - y_{i-2}) : 12) / h; \quad i \geq 2,$$

а при сглаживании по семи точкам — формулой

$$\begin{aligned} y'_i = & ((y_{i+1} - y_{i-1}) \cdot 58 + (y_{i+2} - y_{i-2}) \cdot 67 - \\ & - (y_{i+3} - y_{i-3}) \cdot 22) / 252h; \quad i \geq 3. \end{aligned}$$

Эти формулы реализуются программами 190 и 191 соответственно с занесением шага h в регистр 8 и последовательным вводом отсчетов в регистр X с нажатием клавиши С/П (В/О и С/П для первого пуска).

Программа 190. Вычисление производной со сглаживанием по пяти точкам

$$\begin{array}{rcl} P6 \uparrow F2 - 4 & \div & \rightarrow F5 \uparrow F3 P2 - \\ 2 \times \uparrow F4 P3 & F5 P4 F6 P5 \leftarrow - 3 \\ \div \uparrow F8 \div C/P B\Psi P0 \end{array}$$

Программа 191. Вычисление производной со сглаживанием по семи точкам

$$\begin{array}{rcl} \uparrow \rightarrow F2 - 2 & 2 & \times \rightarrow F7 \uparrow F3 P2 \\ - 6 7 \times \rightarrow F6 \uparrow F4 P3 - 5 & 8 \\ \times \uparrow F5 P4 F6 P5 F7 P6 \leftarrow + & \uparrow \leftarrow \\ - 1 \leftarrow P7 F8 \div 2 5 2 \div C/P B\Psi P0 \end{array}$$

Результат, выводимый на индикатор после ввода i -го отсчета, соответствует производным для $(i-2)$ -го и $(i-3)$ -го узлов при вычислениях по программам 190 и 191, причем в случае необходимости крайних два или три значения производной соответственно вычисляют по формулам, отличающимся от приведенных выше [21].

В вероятностных задачах может возникнуть необходимость в получении последовательности случайных чисел с заданным законом распределения. Для генерирования последовательности квазислучайных чисел с равномерным законом распределения удобно использовать метод Коробова [7], легко программируемый на микрокалькуляторах. Согласно этому методу, очередное число x_{i+1} генерируемой последовательности определяется по формуле

$$x_{i+1} = (qx_i)_{\text{mod}(p)} = [qx_i - pE(qx_i/p)],$$

где p — большое простое число (например, 2027 или 5087), а q подбирается близким к $p/2$ из множества $p = 3^m$, где m — целое число.

Программа 192 ($p = P2$, $q = P3$, $x_0 = PX$, $PX = x_i$) генерирует квазислучайные числа, равномерно распределенные в интервале $[1; p-1]$, которые делением на p легко привести в интервал $[0; 1]$.

Программа 192. Генерирование квазислучайных чисел с равномерным распределением

$$\begin{array}{rcl} \uparrow F3 \times P4 \uparrow F2 \div 1 & B\Psi 7 & XY + \\ XY - \uparrow F2 \times \uparrow F4 XY - & C/P B\Psi P0 \end{array}$$

При $p = 2027$; $q = 1298$ и $x_0 = 1$ по программе 192 последовательно получаем $x_1 = 1298; 367; 21; 907 \dots$

При использовании микрокалькуляторов «Электроника Б3-21» первых выпусков фрагмент программы 192 для выделения целой

части выражения q_{x_i}/p по адресам с 12 по 22 необходимо заменить фрагментом, приведенным в сноске на стр. 224.

Распределение квазислучайных чисел по закону Релея нетрудно получить с помощью преобразования

$$z_i = \sqrt{2\sigma^2 \ln(1/x_i)},$$

в котором используется последовательность случайных чисел x_i с равномерным распределением в интервале $[0; 1]$. По значениям z_i и x_{i-1} можно сформировать последовательность

$$y_i = z_i \sin 2\pi x_{i-1}$$

с нормальным законом распределения. Подобная последовательность генерируется по программе 193 ($p = P2$, $q = P3$, $\sigma^2 = P7$, $x_{i-1} = P6$, $x_i = PX$, $P8 = z_i$, $PX = y_i$).

Программа 193. Генерирование последовательности квазислучайных чисел с нормальным распределением

$$\begin{array}{ccccccccc} \uparrow & F3 & \times & P4 & \uparrow & F2 & \div & 1 & \text{ВП } 7 & XY + \\ XY & - & \uparrow & F2 & \times & \uparrow & F4 & XY - & \uparrow & F6 & XY \\ P6 & F2 & \div & 1/x & \ln & 2 & \times & \uparrow & F7 & \times & \sqrt{P8} \\ F6 & \uparrow & F2 & \div & \uparrow & \pi & \times & 2 & \times & \sin & \uparrow & F8 \\ \times & C/P & F6 & \text{БП } & P0 & & & & & & & \end{array}$$

При исходных данных $p = 2027 = P2$; $q = 1298 = P3$; $\sigma^2 = -1 = P7$, $x_{-1} = 1000 = P6$; $x_0 = 1 = PX$ по программе 193 последовательно получаем $y_1 = -0,9176310$; $0,8569610$; $0,1202592$; $0,9799901 \dots$

8. АВТОМАТИЗАЦИЯ ЗАДАЧ ЛИНЕЙНОГО МОДЕЛИРОВАНИЯ

Большинство физических объектов с достаточной для инженерных расчетов точностью описывается линейными математическими моделями. Процессы накопления потенциальной и кинетической энергий или их аналогов (например, электрической и магнитной энергии) в компонентах физических объектов описывают эквивалентным соединением идеальных дифференцирующих и интегрирующих элементов. С помощью преобразования Лапласа можно отобразить свойства таких идеальных элементов параметрами вида ra или $1/pb$, где $p = \sigma + j\omega$ — комплексная частота (оператор Лапласа), а a и b — вещественные рациональные числа. При этом физическая система со слабыми воздействиями моделируется системой линейных уравнений вида (8), где элементы матрицы коэффициентов и отношения переменных реакции и воздействия, выражаемые через эти элементы формулами (80), являются в общем случае дробно-рациональными функциями комплексной частоты вида (66).

Моделирование физических процессов в области оператора Лапласа позволяет избежать операций с интегрально-дифференциаль-

ными уравнениями. Однако экспериментальное исследование таких процессов при их моделировании функциями комплексной частоты связано со значительными трудностями. Поэтому в инженерной практике широко используется частотный (спектральный) анализ, легко совместимый с результатами экспериментальных исследований, а при необходимости описания естественного протекания процессов их модели, составленные для операторной или частотной областей, преобразуют во временную область.

Переход от дробно-рациональной функции комплексной частоты к ее частотной характеристике упрощается при предварительном выделении в числителе и знаменателе такой функции четных и нечетных частей:

$$F(p) = \frac{(\dots + b_4 p^4 + b_2 p^2 + b_0) + p (\dots + b_5 p^4 + b_3 p^2 + b_1)}{(\dots + a_4 p^4 + a_2 p^2 + a_0) + p (\dots + a_5 p^4 + a_3 p^2 + a_1)}.$$

В этом случае при подстановке $p = j\omega$ получается выражение частотной характеристики

$$F(j\omega) = \frac{(\dots + b'_2 \omega^4 + b'_0 \omega^2 + b_0) + j\omega (\dots + b'_5 \omega^4 + b'_3 \omega^2 + b_1)}{(\dots + a'_4 \omega^4 + a'_2 \omega^2 + a_0) + j\omega (\dots + a'_5 \omega^4 + a'_3 \omega^2 + a_1)},$$

где обозначенные штрихом коэффициенты могут лишь знаком отличаться от соответствующих коэффициентов исходной функции (например, $b'_2 = -b_2$, $b'_4 = b_4$, $b'_3 = -b_3$, $b'_5 = b_5$ и т. д.).

Используя для вычисления многочленов от ω^2 в подобном выражении общую подпрограмму, можно составить программы для автоматического вычисления амплитудно-частотной характеристики $|F(j\omega)|$, и вещественной $\operatorname{Re} F(j\omega)$ и мнимой $\operatorname{Im} F(j\omega)$ частей частотной характеристики, по которым при необходимости определяется фазово-частотная характеристика. При этом для эффективного использования запоминающих устройств микрокалькулятора целесообразно составлять типовые программы для определенных видов заданных дробно-рациональных функций.

Многополюсная функция $BF(p)$, числитель B которой не зависит от p , после нормирования делением на B , группирования четных и нечетных частей числителя и знаменателя и подстановки $p = j\omega$ отображается частотной характеристикой

$$F(j\omega) = \frac{1}{a_6 \omega^6 + a_4 \omega^4 + a_2 \omega^2 + a_0 + j\omega (a_7 \omega^6 + a_5 \omega^4 + a_3 \omega^2 + a_1)}.$$

Амплитудно-частотная характеристика этой функции, а также вещественная и мнимая части ее знаменателя вычисляются по программе 194 ($a_7 = C1$, $a_6 = C2$, $a_5 = C3$, $a_4 = C4$, $a_3 = C5$, $a_2 = C6$, $a_1 = P7$, $a_0 = P8$, $\omega_0 = P2$, $\Delta\omega = P3$) с автоматическим изменением текущего значения частоты $\omega_t = \omega_{t-1} + \Delta\omega$ после каждого выполнения программы. Первый пуск программы (для частоты $\omega = \omega_0$) выполняют нажатием клавишей В/О и С/П, последующие — только клавиши С/П. Текущие значения модуля после каждого выполнения программы заносятся в регистры 4 и X.

вещественной и мнимой частей знаменателя — соответственно в регистры 5 и 6, а текущее значение частоты — в регистр 2.

Программа 194. Вычисление частотной характеристики многополосной функции со степенью знаменателя $m \leq 7$

$$\begin{array}{l}
 F2 \ x^2 \quad P4 \ \Pi \ F/-/ \ F7 + \quad \uparrow \quad F2 \times \quad P6 \leftarrow \\
 \Pi \ F/-/ \rightarrow \ F8 + \quad P5 \ x^2 \quad \uparrow \quad F6 \ x^2 + \checkmark \\
 1/x \ P4 \quad C/\Pi \ F2 \quad \uparrow \quad F3 + \quad P2 \ \Pi \ P0 \leftarrow P5 \\
 \Pi \ P7 \quad \Pi \ F7 \quad \Pi \ F7 \ B/O \ P5 + \quad \uparrow \quad F4 \times \\
 \uparrow \quad F5 \quad \leftarrow \quad \leftarrow \quad B/O
 \end{array}$$

Например, при вычислении амплитудно-частотной характеристики

$$F(j\omega) = \frac{1}{-\omega^6 + 2\omega^4 - 3\omega^2 + 4 + j\omega(-5\omega^6 + 6\omega^4 - 7\omega^2 + 8)}$$

для $\omega_0 = 0$ и $\Delta\omega = 0,5$ по программе 194 после первых пяти пусков получают $|F(j\omega)| = 0,25; 0,2131968; 0,3535533; 0,01936163; 0,002042331$.

В случае необходимости по программе 194 можно организовать вычисление амплитудно-частотной характеристики при изменении частоты по логарифмическому закону. Для этого достаточно ввести в текст программы умножение частоты ω на заданную величину $k = P3$ после каждого обращения к программе. Например, при $k = \sqrt[3]{10}$ на каждую декаду изменения частоты будут вычисляться три значения модуля функции. Если отказаться от автоматического изменения частоты и ограничиться вычислением только модуля функции, то за счет освободившихся регистров памяти можно вычислять амплитудно-частотные характеристики многополосных функций до десятой степени знаменателя. Подобные вычисления реализуются по программе 195 ($a_{10} = C1, a_8 = C2, a_6 = C3, a_4 = C4, a_2 = C5, a_0 = C6, a_9 = P4, a_7 = P5, a_5 = P6, a_3 = P7, a_1 = P8$). Перед каждым пуском программы (клавишами В/О и С/П только первый раз) в регистр индикации вводится значение частоты ω_t .

Программа 195. Вычисление модуля многополосной функции с $m \leq 10$

$$\begin{array}{l}
 x^2 \quad P2 \leftarrow P3 \quad \Pi \ P - \ \Pi \ P - \\
 \Pi \ P - \ \uparrow \quad F3 \leftarrow XY \quad P3 \quad F4 \quad \Pi \ P7 \quad F5 \quad \Pi \ P \\
 PC_x \quad F6 \quad \Pi \ P \ PC_x \quad F7 \quad \Pi \ P \ PC_x \quad F8 + \quad x^2 \quad \Pi \ P7 \\
 F3 \quad x^2 + \checkmark \quad 1/x \quad C/\Pi \quad B/P \quad P0 + \quad \uparrow \quad F2 \times \\
 \uparrow \quad B/O \quad \uparrow \quad F2 \times \quad \uparrow \quad F3 \leftarrow P3 + \quad B/O
 \end{array}$$

Для функции $F(j\omega) = 1/((- \omega^{10} + 4\omega^8 - 8\omega^6 + 6\omega^4 - 2\omega^2 + 10) + j\omega(4\omega^8 - 5\omega^6 + 3\omega^4 - 2\omega^2 + 2))$ при $\omega_t = 0; 1; 2$ по программе 195 соответственно получают $|F(j\omega)| = 0,1; 0,1084652; 0,0006458390$.

Амплитудно-частотная характеристика функций общего вида

$$F(j\omega) = \frac{b_4\omega^4 + b_2\omega^2 + b_0 + j\omega(b_3\omega^2 + b_1)}{a_4\omega^4 + a_2\omega^2 + a_0 + j\omega(a_3\omega^2 + a_1)}$$

вычисляется с помощью программы 196 ($a_4 = C1$, $a_2 = C2$, $a_0 = C3$, $a_3 = C4$, $a_1 = C5$, $b_4 = C6$, $b_2 = P2$, $b_0 = P3$, $b_3 = P4$, $b_1 = P5$) при вводе ω_i в регистр X перед каждым ее пуском (первый пуск — клавишами B/O и C/P).

Программа 196. Вычисление модуля функции общего вида

$$\begin{array}{llllllll} x^2 & P8 & \text{ПП} & 7 & \text{ПП} & C_x & P7 & F6 \\ P9 & F7 & x^2 & + & P7 & F6 & \text{ПП} & 7 \\ F3 & + & P6 & F4 & \text{ПП} & P9 & F5 & + \\ x^2 & + & \uparrow & F7 & \div & V & C/P & B/P \\ P9 & F6 & \leftarrow & P6 & + & B/O & \uparrow & F8 \\ & & & & & & & \times \\ & & & & & & & \uparrow \\ & & & & & & & B/O \end{array}$$

Для функции $F(j\omega) = ((3\omega^4 - 9\omega^2 + 4) + j\omega(-\omega^2 + 5))/(4\omega^4 - 16\omega^2 + 1 + j\omega(-2\omega^2 + 8))$ при $\omega_i = 0; 1; 2; 3$ по программе 196 соответственно вычисляется $|F(j\omega)| = 4; 0,3569152; 16,12451; 0,9071442$.

Если число коэффициентов заданной функции не более 9, то в программу можно ввести операторы автоматического изменения частоты. Например, для вычисления амплитудно-частотной характеристики функции

$$F(j\omega) = \frac{b_2\omega^2 + b_0 + j\omega(b_3\omega^2 + b_1)}{a_4\omega^4 + a_2\omega^2 + a_0 + j\omega(a_3\omega^2 + a_1)}$$

автоматическое изменение частоты на величину $\Delta\omega$ реализуется программой 197 ($a_4 = C1$, $a_2 = C2$, $a_0 = C3$, $a_3 = C4$, $a_1 = C5$, $b_2 = C6$, $b_0 = P2$, $b_3 = P3$, $b_1 = P4$, $\Delta\omega = P7$, $\omega_0 = PX$, $PX = |F(j\omega)|$, $P8 = \omega_i^2$).

Программа 197. Вычисление модуля функции общего вида с приращением частоты

$$\begin{array}{llllllll} x^2 & P8 & \text{ПП} & F7 & \text{ПП} & 7 & P5 & \text{ПП} & P7 \\ x^2 & + & P5 & \text{ПП} & P7 & F2 & + & P6 & F3 \\ + & \text{ПП} & F - & F6 & x^2 & + & \uparrow & F5 & \div \\ V & \uparrow & F7 & + & B/P & P0 & F6 & \leftarrow & P6 \\ \leftarrow & P6 & + & B/O & x^2 & \uparrow & F8 & \times & \uparrow \\ & & & & & & & & B/O \end{array}$$

При $\omega_0 = 0$ и $\Delta\omega = 1$ для функции $F(j\omega) = (-9\omega^2 + 4 + j\omega(-\omega^2 + 5))/(4\omega^4 - 16\omega^2 + 1 + j\omega(-2\omega^2 + 8))$ по программе 197 получают $|F(j\omega)| = 4; 0,5110249; 32,062439; 0,4247546; \dots$

В связи с отсутствием в используемом выходном языке операторов вычисления обратных тригонометрических функций при емкости программной памяти 60 шагов вычисление модуля можно совместить лишь с вычислением тангенс-частотной характеристики $\operatorname{tg}\phi(\omega)$. Только для простейших функций низкого порядка целесообразно

попытаться ввести в программу, кроме вычисления модуля, и вычисление аргумента (фазового угла) частотной характеристики $\varphi(\omega) = \operatorname{arctg} \operatorname{Im} F(j\omega)/\operatorname{Re} F(j\omega)$ с помощью одной из ранее рассмотренных программ для вычисления $\operatorname{arctg} x$.

Обычно фазово-частотную характеристику следует определять на всем интервале $[0; 2\pi]$ или $[-\pi; \pi]$, что требует включения в программу вычисления $\varphi(\omega)$ фрагмента для определения квадранта, в котором расположен искомый угол.

Тангенс-частотная характеристика многополюсной функции со степенью знаменателя $m \leq 6$ вычисляется программой 198 ($a'_6 = C1, a'_4 = C2, a'_2 = C3, a'_0 = C4, a'_5 = C5, a'_3 = C6, a'_1 = P2, \omega_0 = P4, \Delta\omega = P3$) согласно формуле

$$\operatorname{tg} \varphi(\omega) = \frac{\omega(a'_5\omega^4 + a'_3\omega^2 + a'_1)}{a'_6\omega^6 + a'_4\omega^4 + a'_2\omega^2 + a'_0}.$$

Программа 198. Вычисление $\operatorname{tg} \varphi(\omega)$ многополюсной функции

9	0	P7	ПП	F7	ПП	7	ПП	7	P6	F8	ПП
F7	ПП	7	F2	+	↑	F4	×	↑	F6	$x \neq 0$	F/-/
\div	P7	P8	F6	$x < 0$	F/-/	1	8	0	P7	F7	C/П
F4	↑	F3	+	P4	БП	P0	\leftarrow	P8	↑	F4	x^2
\times	↑	F8	\leftarrow	P8	+	B/O					

Выводимый на индикатор результат выполнения программы 198 интерпретируется следующим образом:

- Если высвечено 180, то $\varphi = 180^\circ + \operatorname{arctg} x$, где $x = P8$.
- Если высвечено 90, то $\varphi = \pm 90^\circ$, причем знак φ совпадает со знаком содержимого регистра Y.
- Если высвечивается число, отличающееся от 90 и 180, то это число равно $\operatorname{tg} \varphi$, а $\varphi = \operatorname{arctg} (\operatorname{tg} \varphi)$.

Например, для функции $\operatorname{tg} \varphi(\omega) = \omega(2\omega^4 - 4\omega^2 + 10)/(-\omega^6 + 2\omega^4 - 4\omega^2 + 48)$ при $\omega_0 = 0$ и $\Delta\omega = 1$ после первых пяти выполнений программы 198 получают РХ = 0; 0,1777777; 90 ($РУ > 0$); 180; 180 и Р8 = 0; 0,1777777; 90; -0,7351351; -0,50888. По полученным данным с помощью таблиц или программ вычисления $\operatorname{arctg} x$ получают $\varphi(\omega_i) = 0; 10, 08^\circ; 90^\circ; 143,68^\circ; +153,03^\circ$.

Основной способ построения временных характеристик реакции линейной системы по заданной функции $F(p)$, отображающей реакцию в операторной области (при воздействии единичным импульсом эта функция является входной или передаточной функцией системы, а ее оригинал — импульсной характеристикой), связан с разложением такой функции на простейшие дроби. Для этого необходимо предварительно вычислить корни p_{ni} знаменателя (полюсы функции) и представить функцию в виде

$$F(p) = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_1 p + b_0}{(p - p_{n1})(p - p_{n2}) \dots (p - p_{nn})},$$

где коэффициенты числителя нормированы делением на коэффициент при старшей степени знаменателя.

Ограничимся случаем отсутствия кратных корней, имеющим наибольшее практическое значение. Если все полюсы $F(p)$ вещественны и отсутствует полюс в начале координат (свободный член a_0 знаменателя отличен от нуля), то такая функция раскладывается в простейшую дробь

$$F(p) = \sum_{k=1}^n \frac{A_k}{p - p_{nk}},$$

где коэффициенты A_k (вычты в полюсах) вещественны и вычисляются по формуле

$$A_k = (p - p_{nk}) F(p)|_{p=p_{nk}}.$$

Вычисление вычетов по этой формуле обеспечивается программой 199 ($b_3 = P5$, $b_2 = P6$, $b_1 = P7$, $b_0 = P8$) для функций с чисителем не более третьей степени и знаменателем не более седьмой степени

$$F(p) = \frac{b_3 p^3 + b_2 p^2 + b_1 p + b_0}{(p - p_{n1})(p - p_{n2}) \cdots (p - p_{n7})}.$$

Программа 199. Вычисление вычетов функции с вещественными полюсами

P2	1	P4	F2	↑	←	$x \neq 0$	↑	P3	—	$x \neq 0$	3
↑	F4	×	P4	F2	↑	F3	БП	↑	F5	ПП	ВП
F6	ПП	FBП	F7	ПП	FBП	F8	+	↑	F4	÷	C/П
F2	C/П	БП	P0	+	↑	F2	×	↑	B/O		

После занесения программы 199 в УЗП коэффициенты функции вводятся в ЗУПВ, а значения вещественных полюсов — в произвольном порядке в стек так, что первый из вводимых полюсов оказывается выведенным на индикатор. Если число полюсов меньше семи, то стековорачивается против часовой стрелки так, чтобы один из полюсов высветился на индикаторе. После этого нажимают клавиши В/О и С/П, что приводит к высвечиванию коэффициента A_k . Вычисленное значение A_k регистрируют, а затем нажимают клавишу С/П, что вызывает на индикатор значение полюса, для которого вычислен коэффициент. После этого стек поворачивают до высвечивания следующего полюса и процедура вычислений повторяется. По полученным значениям A_k строят временную характеристику

$$y(t) = \sum_{k=1}^n A_k e^{-p_{nk} t}.$$

Например, для функции $F(p) = (p^3 + 2p^2 + 3p + 4)/(p + 1)(p + 1,5) \times (p + 2)(p + 2,5)(p + 3)$ по программе 199 вычисляют вычеты $A_k = 1,333333; -1,666666; -8; 17,666666; -9,333333$ и строят временную характеристику

$$\dot{y}(t) = 1,333333e^{-t} - 1,666666e^{-1,5t} - 8e^{-2t} + \\ + 17,666666e^{-2,5t} - 9,333333e^{-3t}.$$

Разложение в простейшие дроби функций с полюсом в начале координат обеспечивается программой 200, вычисляющей вычеты функций со знаменателем не более седьмой степени и числителем не более третьей степени.

Программа 200. Вычисление вычетов функции с полюсом в начале координат

P2	1	P4	ПП	FBП	ПП	FBП	ПП	FBП	ПП	FBП	ПП	FBП	ПП
FBП	ПП	FBП	←	F2	$x \neq 0$	÷	↑	F4	×	P4	F5		
ПП	P9	F6	ПП	—	F7	ПП	—	F8	+	↑	F4		
÷	C/П	БП	P0	←	P3	—	$x \neq 0$	F8	↑	F4	×		
P4	F2	↑	F3	B/O	+	↑	F2	×	↑	B/O			

Если степень знаменателя менее семи, то число обращений к подпрограмме по адресу 65 должно быть на единицу меньшим степени знаменателя, а число операторов поворота стека соответственно большим. Чтобы не изменять все адреса обращений к подпрограмме, можно заменить в тексте программы 200 операторами НОП все устранимые операторы. Например, при степени знаменателя 5 число обращений по адресу 65 сокращают до четырех, добавляют два оператора поворота стека после последнего обращения к этой подпрограмме, а устранившиеся операторы заменяют двумя операторами НОП.

При вводе исходных данных для программы 200 отличные от нуля корни знаменателя записывают в стек ($p_1 = C1, p_2 = C2, p_3 = C3, p_4 = C4, p_5 = C5, p_6 = C6$), а коэффициенты числителя b_3, b_2, b_1 и b_0 (допустимо $b_k = 0$) заносят соответственно в регистры 5 — 8 ЗУПВ. Для вычисления вычета в регистр X вводят значение полюса и нажимают клавишу С/П (при первом пуске — клавиши В/О и С/П), что приводит к высвечиванию значения A_k после выполнения программы. В отличие от программы 199 возвращение выбранного полюса в стек происходит автоматически и для вычисления следующего вычета достаточно ввести в регистр X следующий полюс и нажать клавишу С/П.

Например, для вычисления временной характеристики по функции

$$F(p) = (p^3 + 2p^2 + 3p + 4)/p (p + 1) (p + 1,5) (p + 2) (p + 2,5) (p + 3)$$

следует заменить в программе 200 операторы с адреса 20 по 25 фрагментом ПП $\sqrt{ }$ НОП $\leftarrow \leftarrow F2$ и для полюсов РХ = -1;

— 1,5; — 2; — 2,5; — 3; 0 по уточненной программе вычислить

$A_k = -1,333333; 1,111111; 4; -7,066666; 3,111111; 0,1777777.$

В случае комплексных полюсов разложение функции $F(p)$ на простейшие дроби можно выполнить следующим образом:

1. Дифференцируя знаменатель функции по p , составить формулу

$$A_k = B(p)/A'(p) \Big|_{p=p_{nk}}.$$

2. Используя программы для вычисления значений многочленов комплексного аргумента, вычислить значения вычетов при подстановках $p = p_{nk}$.

Подобная методика применима и в случаях, когда все или некоторые полюсы вещественны, а степени числителя и знаменателя больше предусмотренных программами 199 и 200.

При вычислении значений временных (в частности, импульсных и переходных) характеристик для заданного значения t приходится отказаться от использования стека, так как выражения для временных характеристик содержат операторы вычисления экспоненциальной и тригонометрических функций, что может привести к неуправляемому переполнению при работе со стеком. Это ограничивает емкость памяти семью регистрами ЗУПВ, из которых два приходится отводить для хранения текущей переменной и накопления результата.

Если число коэффициентов временной характеристики больше пяти, то целесообразно использовать нормирование как по аргументу (например, $\alpha_i t = x$), так и по одному из коэффициентов. Например, для составления программы вычисления (в некотором интервале изменения t) временной характеристики

$$y(t) = A_1(1 - \alpha_1 t) e^{-\alpha_1 t} + A_2 e^{\alpha_2 t} + A_3 e^{\alpha_3 t} \cos \omega_3 t$$

после замены $\alpha_1 t = x$ и деления правой и левой частей равенства на коэффициент A_1 , получают

$$y(t)/A_1 = (1 - x) e^{-x} + B_2 e^{\beta_2 x} + B_3 e^{\beta_3 x} \cos \gamma_3 x.$$

Эту функцию можно вычислить по программе 201 с исходными данными $\beta_2 = P2$, $\beta_3 = P3$, $\gamma = P4$, $B_2 = P5$, $B_3 = P6$.

Программа 201. Вычисления временной характеристики

P8	↑	F4	×	cos	P7	F3	×	e ^x	↑	F7	×
P7	F8	/—/	1	+	XY	e ^x	×	↑	F7	+	P7
F2	↑	F8	×	e ^x	↑	F5	×	↑	F7	+	C/P
БП	P0										

Если коэффициенты вычисляемой временной характеристики содержат небольшое количество знаков, то их можно записать

непосредственно в программу, что позволит увеличить число членов суммы. Например, значение функции

$$y = 1 + 13e^{-0.5x} - 11e^{-x} + (1 + 3x)e^{-2x} - 4e^{-3x}$$

можно вычислить по программе 202 ($\Delta x = P3$, $x_0 = PX$).

Программа 202. Вычисление временной характеристики с записью коэффициентов в текст

$$\begin{array}{ccccccccc}
 P2 & 3 & \times & 1 & + & P8 & F2 & /-/- & P4 & 2 & \times & ex \\
 \uparrow & F8 & \times & 1 & + & P8 & F4 & 3 & \times & ex & 4 & /-/- \\
 \text{ПП} & P8 & F4 & ex & 1 & 1 & /-/- & \text{ПП} & P8 & F4 & 2 & \div \\
 e^x & 1 & 3 & \text{ПП} & P8 & C/P & F2 & \uparrow & F3 & + & \text{БП} & P0 \\
 \times & \uparrow & F8 & + & P8 & B/O & & & & & &
 \end{array}$$

При $x_0 = 0$ и $\Delta x = 0,25$ по программе 202 последовательно вычисляют $y_i = 0; 3,077613; 4,479751; 5,042303; 5,180417; \dots$

Описанные методы вычисления реакций линейной системы в частотной и временной областях по операторному представлению функций применимы в случаях, когда вычисляется импульсная или переходная характеристика или когда воздействие с достаточной точностью можно описать в операторной форме. Однако при воздействии произвольной или значительно отличающейся от представимой по Лапласу формы составление операторного выражения для реакции может оказаться чрезмерно сложной задачей. В подобных случаях для анализа линейных систем целесообразно воспользоваться методами цифрового моделирования.

Пример 69. Составить программу для определения изменения температуры корпуса транзистора при усилении сигналов сложной формы.

Связь между температурой T_2 корпуса транзистора и тепловым потоком $P = kI_k$, определяемым мощностью рассеяния на коллекторе и пропорциональным току коллектора I_k , приближенно моделируется эквивалентной схемой, показанной на рис. 43, а. В этой схеме R_1 и R_2 — тепловые сопротивления между областью коллектора и корпусом и между корпусом и окружающей средой соответственно, а C_1 и C_2 — теплоемкости области коллектора и корпуса соответственно.

Предполагая, что до подачи сигнала на вход транзистора температуры области коллектора T_1 и корпуса T_2 практически не отличаются от температуры окружающей среды (если это предположение неточно, то можно учесть постоянную составляющую теплового потока), составим для эквивалентной схемы систему линейных уравнений равновесия в операторной форме и найдем выражение передаточной функции

$$K(p) = \frac{T_2}{kI_k} = \frac{\frac{1}{R_1C_1C_2}}{p^2 + p\left(\frac{1}{R_1C_1} + \frac{R_1 + R_2}{R_1R_2C_2}\right) + \frac{1}{R_1R_2C_1C_2}} = \frac{K_0}{p^2 + p\omega_0d + \omega_0^2},$$

связывающей температуру корпуса T_2 с током I_k коллектора.

В соответствии с выражениями (78) находим передаточную z -функцию

$$K(z) = \frac{K_0(1 + 2z^{-1} + z^{-2})}{1 + \omega_0d + \omega_0^2 + 2(\omega_0^2 - 1)z^{-1} + (1 - \omega_0d + \omega_0^2)z^{-2}}$$

и разностное уравнение

$$y(i) = \frac{x(i) + 2x(i-1) + x(i-2) - W_1 y(i-1) - W_2 y(i-2)}{W_0}.$$

Дополнив программу 88 фрагментом, учитывающим задержку на один такт входного воздействия с весом 2, получим для вычисления температуры корпуса транзистора программу 203 ($W_0 = P2$, $W_1/W_2 = P3$, $-W_2 = P4$, $x_i = PX$, $PX = y_i$).

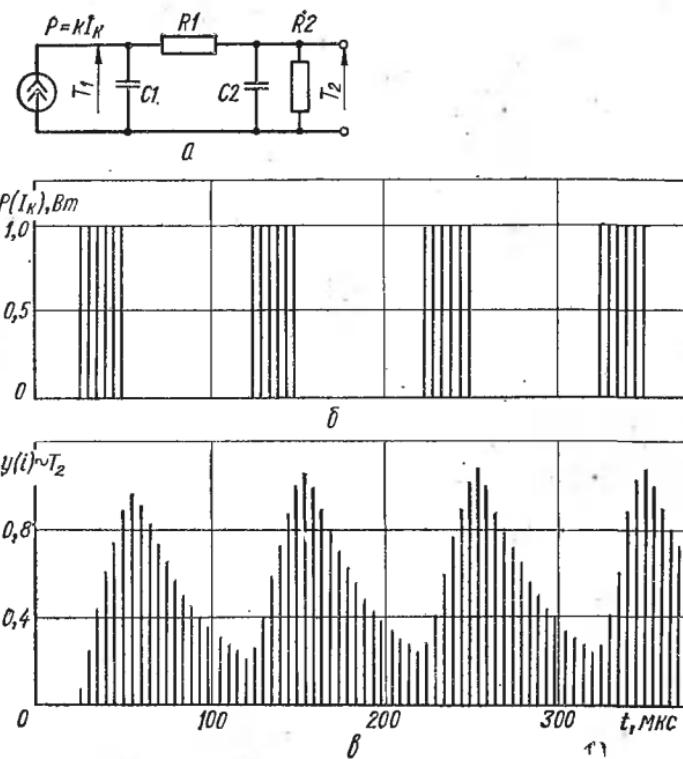


Рис. 43. Цепь теплопередачи от коллектора к корпусу:
а — эквивалентная схема; б, в — график воздействия и реакции

Программа 203. Вычисление температуры корпуса транзистора

\uparrow	\leftarrow	P8	XY	$+$	$+$	\uparrow	F8	\leftarrow	$+$	\leftarrow	\uparrow
F3	\times	XY	\leftarrow	$+$	\uparrow	F4	\times	\uparrow	\rightarrow	\rightarrow	$+$
\uparrow	F2	\div	\uparrow	\rightarrow	\rightarrow	\rightarrow	XY	C/P	BП	P0	

Пусть нормированные параметры исходной модели $R_1 = 0,5$, $R_2 = 2$, $C_1 = 10^{-2}$, $C_2 = 10^{-2}$. Выберем шаг дискретизации $\theta = 5$ мс и согласно соотношению (144) для $\omega_{\text{ц}} = 10^2$ вычислим $\omega_A = \operatorname{tg} \omega_{\text{ц}} \theta / 2 = 0,2553419$. Тогда $W_1 = 2(\omega_{0A}^2 - 1)/(R_2 \omega_{0A}^2) = -14,33754$; $W_2 = (1 - \omega_{0A} d + \omega_{0A}^2)/(R_2 \omega_{0A}^2) = -0,6429422$; $W_0 = (1 + \omega_{0A} d + \omega_{0A}^2)/(R_2 \omega_{0A}^2) = 16,98048$.

Предположим, что коллекторный ток транзистора при усилении сигнала образован последовательностью прямоугольных импульсов длительностью $\tau = 25$ мс с паузами длительностью 75 мс, а максимальное значение тока

соответствует мгновенной мощности рассеивания 1 Вт. Для выбранного шага дискретизации такое воздействие моделируется решетчатой функцией, показанной на рис. 43, б. При вводе отсчетов воздействия в регистр индикации и вычислении реакции T_2C по программе 203 получим решетчатую функцию (рис. 43, в), численно характеризующую процессы изменения температуры корпуса транзистора.

В примере вычисления по программе 203 воздействие имеет часто встречающуюся форму прямоугольных импульсов, но составленная программа пригодна и для любой другой формы воздействия.

В случае периодических воздействий вычисления при цифровом моделировании можно упростить, дополняя программу фрагментом автоматического формирования следующего значения воздействия после каждого выполнения программы. Подобные фрагменты можно использовать и в качестве самостоятельных программ генерирования табличных моделей заданных периодических или непериодических функций. Примером может служить программа 204 для генерирования табличной модели (с шагом θ) последовательности прямоугольных импульсов по данным примера 68 с $T = 20\theta$, $\tau = 5\theta$.

Программа 204. Генерирование табличной модели последовательности прямоугольных импульсов с $\tau/\theta = 5$, $T/\theta = 20$

F5	1	+	P5	2	0	-	$x = 0$	FXY	P5	1	4
+	$x \geq 0$	P3	C _x	БП	F3	F7	C/P	БП	P0		

Перед пуском этой программы нажатием клавиш В/О и С/П (для последующих пусков нажимают только клавишу С/П) регистр 5 очищают, а в регистр 7 вводят максимальное значение импульса. При использовании этой программы совместно с программой 203 три последних оператора программы 204 исключают и вписывают остальную ее часть в начале программы 203.

Сравнительно просто реализуется программное генерирование табличных моделей и более сложных функций, в частности, генерирование табличной модели гармонических воздействий вида $x_i = X_m \sin 2\pi i\theta/T$ или $x_i = X_m \cos 2\pi i\theta/T$. В первом случае программа имеет вид

F8	sin	↑	F2	×	C/P	F8	↑	F3	+	P8	БП	P0
----	-----	---	----	---	-----	----	---	----	---	----	----	----

с исходными данными $X_m = P2$, $2\pi\theta/T = P3$, $0 = P8$. Для генерирования табличной модели косинусоиды достаточно заменить в такой программе второй оператор оператором cos.

Программы генерирования табличных моделей сложных функций имеют большее число операторов. Например, последовательность усеченных косинусоидальных импульсов (типичная для детектирующих или выпрямительных устройств), отображаемая табличной моделью $x_i = x_{\max} (\cos 2\pi i\theta/T - \cos \psi)$, при $(\cos 2\pi i\theta/T - \cos \psi) \geq 0$ и $x := 0$ при $(\cos 2\pi i\theta/T - \cos \psi) < 0$ генерируется по программе 205 ($2\pi\theta/T = P5$, $\cos \psi = P6$) при $x_{\max} = 1$.

Программа 205. Генерирование табличной модели последовательности усеченных косинусоидальных импульсов

F7 cos ↑ F6 — $x < 0$ PX Y 0 C/P F7 ↑ F5
+ P7 БП Р0

При $2\pi\theta/T = \pi/5 = 0,6283185 = P5$, $\cos\psi = -0,5 = P6$ по программе 205 получают $x_t = 1,5; 1,309017; 0,809017; 0,1909831; 0; 0; 0,1909828; 0,8090166; 1,309016; 1,5; \dots$

Если $x_{max} \neq 1$, то программа 205 дополняется фрагментом ↑ P8 × перед оператором С/П с предварительной записью $x_{max} = P8$. В случае, когда программа 205 используется для автоматического формирования входной последовательности цифрового фильтра, программа последнего вписывается в программу 205, начиная с оператора по адресу 13, а соответствующая часть программы 205 записывается в конце текста объединенной программы. В линейном фильтре целесообразно принять $x_{max} = 1$ с целью сокращения длины программы.

При анализе дискретных систем или каналов с цифровой обработкой, содержащих на выходе интерполирующий фильтр, может возникнуть необходимость в интерполяции заданной последовательности отсчетов непрерывной функции времени. Возможность автоматизации этой операции в основном определяется емкостью его запоминающих устройств. Например, при интерполяции временным рядом Котельникова

$$f(t) = \sum_{k=-\infty}^{\infty} f(k\Delta t) \frac{\sin \omega_b(t - k\Delta t)}{\omega_b(t - k\Delta t)}$$

число отсчетов, одновременно хранимых в запоминающих устройствах микрокалькулятора типа «Электроника Б3-21», ограничено пятью регистрами ЗУПВ, так как стек нельзя использовать в связи с вычислениями функции $\sin x$. При решении подобной задачи вычисления целесообразно выполнять над нормированными значениями переменных

$$f(x) = \sum_{k=-\infty}^{\infty} f(k) \frac{\sin \pi(x - k)}{\pi(x - k)},$$

где $x = t/\Delta t$.

Сумму членов ряда Котельникова при таком нормировании можно вычислить по программе 206 ($f(0) = P2$, $f(1) = P3$, $f(2) = P4$, $f(3) = P5$, $f(4) = P6$, $x = PX$, $PX = f(x)$). Время счета около 25 с.

Программа 206. Вычисление суммы членов ряда Котельникова

P8 0 P7 F8 ПП /—/ F2 ПП → F3 ПП →
F4 ПП → F5 ПП → F6 × ↑ F7 + — P8 ↑
БП Р0 × ↑ F7 + P7 F8 1 — P8 ↑
 π × sin XY ÷ ↑ B/O

Например, при $f(0) = f(1) = f(2) = f(3) = f(4) = 1$ по программе 206 получают $f(-0,5) = 0,5315269$; $f(0,5) = 1,097411$; $f(1,5) = 0,9761502$ и т. д. Заметим, что при целом значении аргумента x вычисления по этой программе невозможны в связи с переполнением, но в соответствии с приведенной выше формулой в этом случае результат равен $f(x = k)$.

Следует добавить, что необходимым условием линейности математической модели является устойчивость моделируемого физического объекта, при которой все корни характеристического уравнения $\Delta(p) = 0$ имеют отрицательные вещественные части. В физических устройствах, содержащих внутренние источники энергии, возможно самовозбуждение, в связи с чем исследование устойчивости таких устройств по их математическим моделям является неотъемлемым этапом инженерного проектирования.

При исследовании условий устойчивости для вычисления корней определителя матрицы коэффициентов системы уравнений, моделирующей физический объект, можно воспользоваться программами вычисления корней алгебраических многочленов степени $m \leq 6$. При большей степени определителя $\Delta(p)$ вычисление его корней на микрокалькуляторах связано с большими затратами времени.

Задача упрощается, если требуется только оценить устойчивость, т. е. определить, устойчив или неустойчив моделируемый объект. В этом случае иногда используют критерий Рауса — Гурвица, требующий вычисления ряда определителей для проверки условий (65а). При порядке вычисляемых определителей $m > 3$ оценка устойчивости по критерию Рауса — Гурвица становится громоздкой и более удобны частотные критерии.

Относительно двух входов, охваченных петлей обратной связи (см. рис. 1, б), физический объект при слабых воздействиях описывается линейной системой уравнений вида

$$\begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}.$$

Определитель матрицы коэффициентов такой системы

$$\Delta = W_{11}W_{22} - W_{12}W_{21} = W_{11}W_{22}(1 - \mu\beta),$$

где прямая передача $\mu = \Delta_{12}/\Delta_{11} = -W_{21}/W_{22}$ и обратная передача $\beta = \Delta_{21}/\Delta_{22} = -W_{12}/W_{22}$.

Если корни многочлена $W_{11}W_{22}$ заведомо расположены на плоскости Лапласа слева от оси $j\omega$, то задача исследования устойчивости сводится к анализу расположения корней множителя $1 - \mu\beta$ или его составляющей $\mu\beta$. На этом основан используемый в инженерной практике критерий Найквиста. Согласно этому критерию, система с обратной связью устойчива, если частотный годограф $\mu\beta$ не охватывает критической точки с координатами $(1; 0)$ или годограф $1 - \mu\beta$ не охватывает начала координат. В тех случаях, когда расположение корней множителя $W_{11}W_{22}$ точно не известно, но они могут

располагаться на оси $j\omega$ или справа от нее, критерий Найквиста недостаточен и его необходимо заменить более общим критерием (65в). В обоих случаях задача сводится к построению частотного годографа алгебраического многочлена с аргументом ω .

Построение частотного годографа многочлена степени $m \leq 10$ обеспечивается программой 207, перед использованием которой исследуемый многочлен $P(p)$ представляется посредством подстановки $p = j\omega$ в виде

$$P(j\omega) = a_{10}\omega^{10} + a_8\omega^8 + a_6\omega^6 + a_4\omega^4 + a_2\omega^2 + a_0 + j\omega(a_9\omega^8 + a_7\omega^6 + a_5\omega^4 + a_3\omega^2 + a_1).$$

Программа 207. Построение частотной характеристики многочлена степени $m \leq 10$

x^2	P2	\leftarrow	P3	ПП	F8	ПП	F8	ПП	F8	ПП	F8
ПП	F8	\uparrow	F3	\leftarrow	XY	P3	F4	ПП	PC _x	F5	ПП
F7	F6	ПП	F7	F7	ПП	F7	F8	+	\uparrow	F2	V
P2	\times	\uparrow	F3	C/P	BП	F0	+	\uparrow	F2	\times	\uparrow
B/O	\uparrow	F2	\times	\uparrow	F3	\leftarrow	P3	+	B/O		

Исходные данные для программы 207 включают значение частоты $\omega = PX$ и коэффициенты многочлена $a_{10} = C1$, $a_8 = C2$, $a_6 = C3$, $a_4 = C4$, $a_2 = C5$, $a_0 = C6$, $a_9 = P4$, $a_7 = P5$, $a_5 = P6$, $a_3 = P7$, $a_1 = P8$. После ввода начального значения частоты нажимают клавиши B/O и C/P, после ввода последующих значений — только клавишу C/P. По окончании выполнения программы на индикаторе высвечивается $ReP(j\omega)$, а для вызова $ImP(j\omega)$ достаточно нажать клавишу X ↔ Y. Значение x сохраняется для справок в регистре 2 ЗУПВ.

Частотный годограф многочлена одиннадцатой степени можно построить с помощью программы 208 с тем же размещением исходных данных, что и для предыдущей программы. Для использования программы 208 многочлен степени $m = 11$ нормируется делением на коэффициент a_{11} (так как обычно $a_{11} < 0$, то в этом случае знаки остальных коэффициентов изменяются на обратные).

Программа 208. Построение частотной характеристики многочлена степени $m = 11$

x^2	P2	\leftarrow	P3	ПП	8	ПП	8	ПП	8	ПП	8
ПП	8	\uparrow	F3	\leftarrow	XY	P3	F2	\uparrow	F4	ПП	7
F5	ПП	7	F6	ПП	7	F7	ПП	7	F8	+	\uparrow
F2	V	P2	\times	\uparrow	F3	C/P	BП	P0	+	\uparrow	F2
\times	\uparrow	B/O	\uparrow	F2	\times	\uparrow	F3	\leftarrow	P3	+	B/O

При построении частотных годографов с помощью программ 207 и 208 целесообразно принять начальное значение частоты $\omega_0 = 0$ с достаточно большим шагом, чтобы получить представление о форме годографа. Если он проходит вблизи критической точки, то в этом случае следует повторить вычисления с меньшим

шагом вблизи критической точки. Так как ветви годографа с $\omega < 0$ и $\omega > 0$ симметричны, то достаточно выполнить вычисления при $\omega > 0$.

При $m \leq 7$ для быстрой оценки устойчивости многочленов целесообразно использовать критерий (65б), из которого непосредственно вытекает и критерий Рауса — Гурвица. Оценка устойчивости по критерию Гурвица для многочленов степени 4—7 выполняется с помощью программ 209—212 соответственно. Результат выполнения этих программ выводится на индикатор в виде положительного числа (равного последнему элементу разложения в цепную дробь), если многочлен устойчив. При неустойчивости многочлена результат отрицателен (и равен первому отрицательному коэффициенту разложения) или возникает переполнение (при равенстве нулю одного из коэффициентов).

Исходными данными для программы 209 являются коэффициенты многочлена $P(p)$ четвертой степени $a_4 = P2, a_3 = P3, a_2 = P4, a_1 = P5, a_0 = P6$. Время счета около 5 с.

Программа 209. Оценка устойчивости многочлена четвертой степени

$$\begin{array}{ccccccccc} F2 & \uparrow & F5 & \times & \uparrow & F3 & \div & \uparrow & F4 - / \overline{-} / \\ F4 & P4 & F3 & \uparrow & F6 & \times & \uparrow & F4 \div & \uparrow F5 - \\ / \overline{-} / & C/P & . & . & . & . & . & . & . \end{array} \quad x \geq 0$$

Для многочлена $2p^4 + 3p^3 + 4p^2 + 5p + 1$ по программе 209 получают $PX = 0,5000003$ (многочлен устойчив), для многочлена $2p^4 + 3p^3 + p^2 + 5p + 1$ — соответственно $PX = -2,333333$ (многочлен неустойчив).

Для оценки устойчивости многочлена пятой степени используется программа 210 (время счета около 12 с) с исходными данными $a_5 = C1, a_4 = C2, a_3 = C3, a_2 = C4, a_1 = C5, a_0 = C6$.

Программа 210. Оценка устойчивости многочлена пятой степени

$$\begin{array}{ccccccccc} \leftarrow & P4 & \Pi\Pi & P3 & \uparrow & F4 & XY & \leftarrow & \Pi\Pi & 3 & \rightarrow & \rightarrow \\ \Pi\Pi & \times & \Pi\Pi & \times & C/P & \rightarrow & \uparrow & \leftarrow & P2 & \leftarrow & \leftarrow & P3 \\ \times & \uparrow & F2 & \div & \uparrow & F3 & \rightarrow & XY & - & x \geq 0 & F \times & B/O \end{array}$$

При оценке устойчивости многочлена $12p^5 + 12p^4 + 16p^3 + 10p^2 + 5p + 1$ по программе 210 получают $PX = 1$ (многочлен устойчив), для $p^5 + 3p^4 + 12p^3 + p^2 + 6p + 2$ — соответственно $PX = -0,3714284$ (многочлен неустойчив), для $p^5 + 2p^4 + 2p^3 + 4p^2 + p + 1$ возникает переполнение (многочлен неустойчив).

Для оценки устойчивости многочлена шестой степени предназначена программа 211 (время счета около 20 с) с исходными данными $a_5 = C1, a_4 = C2, a_3 = C3, a_2 = C4, a_1 = C5, a_0 = C6, a_6 = = PX$.

Программа 211. Оценка устойчивости многочлена шестой степени

ПП F4 ↑ F4 XY ←	ПП , ПП F ÷ →	ПП F4 C/P ↑ F4
XY ← ПП , ПП F ÷ →	ПП F4 C/P ↑ F4	ПП F4 C/P ↑ F4
→ P4 ↑ ← P2 ← ← P3 × ↑ F2 ÷	→ P3 × ↑ F2 ÷	
↑ F3 → XY — x ≥ 0 3 B/O		

При оценке устойчивости многочлена $p^6 + 21p^5 + 175p^4 + 735p^3 + 1624p^2 + 1764p + 720$ по программе 211 получают $PX = 1320$ (многочлен устойчив), для $2p^6 + 4p^5 + 2,5p^4 + p^3 + 6p^2 + 8p + 12$ — соответственно $PX = -3$ (многочлен неустойчив).

Для оценки устойчивости многочлена седьмой степени по программе 212 (время счета около 30 с) он нормируется делением на коэффициент при старшей степени и в регистры памяти заносятся нормированные коэффициенты $a'_6 = P2$, $a'_5 = C1$, $a'_4 = C2$, $a'_3 = C3$, $a'_2 = C4$, $a'_1 = C5$, $a'_0 = C6$.

Программа 212. Оценка устойчивости нормированного многочлена седьмой степени

ПП ВП ПП ВП ПП ВП ← F4 ПП РВП ПП ВП	ПП F6 ПП ВП
← P3 F4 ← F3 ↑ F2 ÷ ПП F6 ПП ВП	ПП F4 ПП ВП
← F4 ← ПП F6 ↑ F2 × P2 F4 ПП ВП	P7
C/P ← ← ← P2 ← P3 F4 ← ↑ F2 ÷	
P4 F3 ↑ F4 — ↑ F2 ÷ x ≥ 0 P6 B/O	

При оценке устойчивости многочлена $p^7 + 21p^6 + 196p^5 + 910p^4 + 2359p^3 + 3388p^2 + 2484p + 720$ по программе 212 получают $PX = 0,3610711$ (многочлен устойчив).

Если в приведенные выше программы для разложения в цепные дроби ввести оператор остановки С/П после каждого деления, то их можно использовать, в частности, для синтеза двухполюсников по канонической схеме Кауэра [16]. Примером может служить программа 213 для деления двух многочленов до пятой степени (или до десятой, если все степени четные) с перезаписью делится на место делимого, а остатка — на место делителя. Перед выполнением программы коэффициенты делимого и делителя записываются соответственно в стек и ЗУПВ в порядке убывания или возрастания их номеров. Очередной результат деления высвечивается.

Программа 213. Разложение дробно-рациональной функции в цепную дробь

F2 ↑ ← XY ÷ P8 F3 ПП F — F3 ← XY	
— P2 F4 ПП F — F4 ← XY — P3 F5 ПП	
F — F5 ← XY — P4 F6 ПП F — F6 ← XY	
— P5 F7 ПП F — F7 ← XY — P6 0 P7	
F8 C/P БП P0 ↑ F8 × ↑ B/O	

Пример 70. Синтезировать двухполюсник с входной проводимостью

$$Y(p) = \frac{3 \cdot 10^{-4} p^5 + 1,44 p^3 + 880 p}{0,03 p^4 + 132 p^2 + 4 \cdot 10^4}.$$

Заносим в УЗП программу 213 и вводим коэффициенты заданной функции входной проводимости следующим образом: $3 \cdot 10^{-4} = C1$; $4,65 = C2$; $65 \cdot 10^2 = C3$; $0,03 = P2$; $450 = P3$; $5 \cdot 10^5 = P4$. После нажатия клавишей В/О и С/П получаем результат первого деления $RX = 0,01$. Повторно нажимая клавишу С/П, последовательно получаем $RX = 0,25; 0,01; 0,15; 0,002$. Так как в канонической схеме Кауэра первым элементом является параллельная емкость, то следующим будет последовательная индуктивность. Следовательно, по вычисленным результатам получим схему замещения двухполюсника (рис. 44) с заданной входной проводимостью.

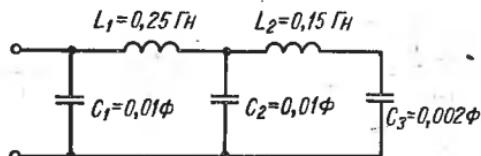


Рис. 44. Схема синтезированной электрической цепи

шего при сокращении на 2 степени остатка. В таких случаях следует определить коэффициенты многочленов, вычисленных до переполнения или получения отрицательного результата, и, перезаписав их в регистры памяти для другого способа деления, продолжить вычисления.

При синтезе цепей с потерями методами, связанными с разложениями в цепные дроби, иногда приходится изменять порядок деления многочленов числителя и знаменателя заданной функции в случае получения отрицательных коэффициентов или переполнения, возникаю-

9. ОСОБЕННОСТИ АВТОМАТИЗАЦИИ НЕЛИНЕЙНОГО МОДЕЛИРОВАНИЯ

Нелинейное моделирование вследствие зависимости свойств моделируемого физического объекта от воздействий неизмеримо сложнее линейного. Значительные трудности, связанные с проблемой адекватности физического объекта его математической модели, возникают еще на этапе составления модели, повторяясь при изменении условий работы. Аналитическое определение по составленной модели свойств нелинейных объектов при заданных воздействиях оказывается возможным лишь в исключительных случаях и основными являются численные (итерационные) методы. При этом существенное значение приобретает проверка сходимости решения к имеющим физический смысл результатам, так как методическая погрешность может привести не только к численным ошибкам (как в линейном моделировании), но и к принципиально неверным результатам, не соответствующим физическому состоянию моделируемого объекта.

Сложность нелинейного моделирования ограничивает использование программируемых микрокалькуляторов с малой емкостью запоминающих устройств расчетами, связанными с наиболее простыми моделями. Однако и в этом случае для размещения исходных данных и программы в запоминающих устройствах обычно приходится прибегать к упрощениям модели, допустимым из физи-

ческих соображений, ее преобразованию с нормированием переменных и параметров и тщательно выбирать алгоритм решения.

Рассмотрим для примера моделирование цепей постоянного тока с биполярными транзисторами. В статическом режиме транзистор типа $p-n-p$ обычно приближенно описывают безынерционной моделью Эберса — Молла

$$\left. \begin{aligned} I_9 &= \frac{I_{90}}{1 - \alpha_N \alpha_I} (e^{\Lambda U_{9b} - \Lambda r'_b I_b} - 1) - \frac{\alpha_I I_{k0}}{1 - \alpha_N \alpha_I} (e^{\Lambda U_{kb} - \Lambda r'_b I_b} - 1); \\ I_k &= -\frac{\alpha_N I_{90}}{1 - \alpha_N \alpha_I} (e^{\Lambda U_{9b} - \Lambda r'_b I_b} - 1) + \frac{I_{k0}}{1 - \alpha_N \alpha_I} (e^{\Lambda U_{kb} - \Lambda r'_b I_b} - 1), \end{aligned} \right\} \quad (149a)$$

где положительными приняты направления токов, втекающих в транзистор, а знаки напряжений учитываются при подстановке численных значений.

Для транзисторов типа $n-p-n$ положительные направления токов и знаки показателей экспонент изменяются на противоположные. Значения параметров модели зависят от типа транзистора (например, для кремниевых маломощных транзисторов типовые значения $I_{90} \approx I_{k0} = 10 \text{ мкА}$; $\alpha_N = 0,995$; $\alpha_I = 0,75$; $\Lambda = 40 \text{ 1/B}$; $r'_b = 120 \text{ Ом}$).

В зависимости от режима работы транзистора приведенную модель можно упростить. В частности, в нормальном режиме работы $|U_{kb}| > 0,3 \text{ В}$ и соответствующий экспоненциальный член оказывается пренебрежимо малым по сравнению с единицей, что позволяет привести модель к виду

$$I_9 = A_{9,9} (e^{\Lambda U_{9b} - \Lambda r'_b I_b} - 1) + A_{9k}; \quad I_k = A_{k9} (e^{\Lambda U_{9b} - \Lambda r'_b I_b} - 1) + A_{kk}, \quad (149b)$$

где значения коэффициентов A_{ij} очевидны из приведенных ранее формул.

В связи с малостью тока базы $I_b = -(I_k + I_9)$ часто принимают $r'_b = 0$, что еще больше упрощает модель транзистора ($A_{9,k} \approx A_{kk} \approx 0$)

$$I_9 = A_{9,9} (e^{\Lambda U_{9b}} - 1); \quad I_k = -A_{k9} (e^{\Lambda U_{9b}} - 1), \quad (149b)$$

откуда, в частности,

$$I_b \equiv -(I_9 + I_k) = -\frac{(1 - \alpha_N) I_{90}}{1 - \alpha_N \alpha_I} (e^{\Lambda U_{9b}} - 1).$$

Безынерционные модели транзистора в основном используются при расчете цепей постоянного тока, когда можно полностью пре-небречь инерционными свойствами элементов. При этом приходится решать как задачу синтеза, заключающуюся в определении сопротивлений линейных элементов схемы, обеспечивающих требуемый режим питания транзисторов (определенный двумя токами или напряжениями на их выводах) при заданных напряжениях источников питания, так и задачу анализа, заключающуюся в определении ре-

жима питания (рабочей точки) транзисторов при заданных сопротивлениях остальных элементов цепи и напряжениях источников питания.

Задача синтеза цепи постоянного тока сводится к составлению и решению системы линейных относительно сопротивлений элементов уравнений и сравнительно проста. Значительно более сложна задача анализа, требующая решения системы нелинейных уравнений.

Рассмотрим распространенную задачу определения рабочей точки транзистора в схеме, содержащей в общем случае шесть резисто-

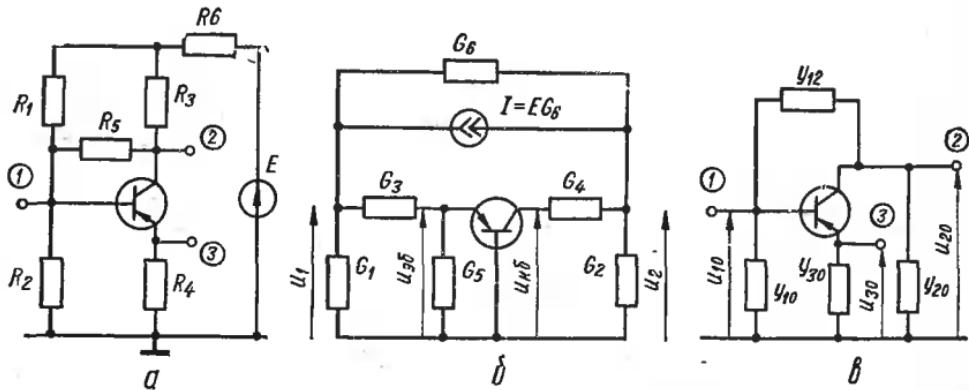


Рис. 45. Обобщенные схемы транзисторной ступени

ров (рис. 45, а). Размыкая или закорачивая резисторы этой схемы, можно получить практически все используемые на практике схемы постоянного тока транзисторной ступени.

Выбрав вывод базы в качестве общего узла и преобразовав источник питания, получим эквивалентную схему, показанную на рис. 45, б, для которой составим систему уравнений узловых напряжений (где G_i — проводимости резисторов схемы с идентичными индексами)

$$\begin{aligned} -G_6 E &= (G_1 + G_3 + G_6) U_1 - G_6 U_2 - G_3 U_{96}; \\ G_6 E &= (G_2 + G_4 + G_6) U_2 - G_6 U_1 - G_4 U_{96}; \\ I_9 &= (U_2 - U_{96}) G_4; \\ I_K &= G_3 U_1 - (G_3 + G_5) U_{96}. \end{aligned}$$

Исключив переменные U_1 и U_2 и вводя обозначения

$$\begin{aligned} a &= 1 + G_1 R_3; \quad b = 1 + G_2 R_4; \quad c = 1 + G_2 R_6; \\ d &= 1 + G_1 R_6; \quad q = aR_4 + bR_3 + abR_6; \\ f &= G_1 R_4 + bd + G_5 q; \quad h = G_2 R_3 + ac, \end{aligned}$$

получим систему уравнений

$$qI_9 = aE + U_{96} - hU_{96}; \quad qI_K = -bE + U_{96} - fU_{96}.$$

Приняв те же допущения, что и для системы уравнений (149в), можно подставить в полученную систему уравнений $I_9 = I_1(e^{\Delta U_{96}} - 1)$, $I_k = -\alpha_N I_9$ и, исключив $U_{kб}$, получить нелинейное уравнение

$$I_1(e^{\Delta U_{96}} - 1) = [E(f - b) + U_{96}(1 - fh)]/q(f - \alpha_N)$$

или, после его преобразования для обеспечения сходимости,

$$U_{96} = \frac{1}{\Lambda} \ln(AE + BU_{96} + 1),$$

где

$$A = (af - b)/qI_1(f - \alpha_N); \quad B = (1 - fh)/qI_1(f - \alpha_N).$$

Решив полученное уравнение, например, методом простых итераций, напряжение $U_{kб}$ можно определить по уравнению

$$U_{kб} = CE + DU_{96},$$

где

$$C = (\alpha_N a - b)/(f - \alpha_N); \quad D = (1 - \alpha_N h)/(f - \alpha_N).$$

Заметим, что одновременно можно определить слабосигнальные низкочастотные параметры транзистора в рабочей точке — передаточную проводимость (крутизну коллекторно-базовой статической характеристики) $g_{kб} = S = dI_k/dU_{69} = \Lambda \alpha_N I_9$ и входную проводимость транзистора $g_{бб} = dI_b/dU_{69} \approx \Lambda I_9(1 - \alpha_N)$, используя которые, несложно вычислить коэффициент усиления напряжения для слабых сигналов [24]

$$K = \frac{U_{k0}}{U_{60}} = \frac{-(g_{kб} - y_{12}) + y_{12}(g_{kб} + g_{бб})/y_{30}}{y_{12} + y_{20} + (g_{kб} + g_{бб})(y_{20} + y_{12})/y_{30}}$$

и входную проводимость ступени

$$y_{вх} = y_{10} + \frac{y_{20}(g_{бб} + y_{12}) + y_{12}(g_{бб} + g_{kб})(1 + y_{20}/y_{30})}{(g_{бб} + g_{kб} + y_{20})(1 + y_{12}/y_{30})},$$

где y_{ij} — полные проводимости между узлами i и j эквивалентной схемы транзисторной ступени для слабых сигналов (рис. 45, в).

Таким образом, решение задачи анализа транзисторной ступени на постоянном токе и низкой частоте включает вычисление коэффициентов $a, b, c, d, q, f, h, A, B, C$ и D (при схеме питания, отличающейся от показанной на рис. 45, а, достаточно в последней принять равными нулю сопротивления закорачиваемых и проводимости размыкаемых резисторов) по заданным параметрам схемы, решить нелинейное уравнение относительно U_{96} и вычислить по приведенным выше формулам величины $U_{kб}, I_9, g_{kб}, g_{бб}$.

Последние два этапа выполняются автоматически при использовании программы 214 с исходными данными $AE + 1 = P2, B = P3, CE = P4, D = P5, \alpha_N = P6, I_1 = P7$, причем принимается $E = |E|$.

Программа 214. Определение режима питания и слабосигнальных параметров цепи с биполярным транзистором

$$\begin{array}{ccccccccc}
 & F3 & \times & \uparrow & F2 & + & \ln & 3 & 9 \\
 \uparrow & XY & P8 & - & x=0 & F0 & F5 & XY & C/P \\
 X/Y & P8 & 3 & 9 & \times & e^x & 1 & - & \uparrow \\
 C/P & F8 & 3 & 9 & \times & e^x & 1 & - & \uparrow \\
 3 & 9 & \times & P8 & \uparrow & F6 & \times & C/P & F6 \\
 \uparrow & F8 & \times & C/P & & & & & 1 \\
 & & & & & & & & XY \\
 & & & & & & & & -
 \end{array}$$

Перед пуском программы по команде В/О, С/П в регистр индикации вводится начальное значение $U_{кб} < E$. После выполнения программы (время счета около 30 с) на индикаторе высвечивается значение $U_{вб}$. После каждого последующего нажатия клавиши С/П поочередно вычисляются значения $U_{кб}$, I_s , $g_{кб}$ и $g_{бб}$. Знаки выводимых на индикатор результатов соответствуют транзистору типа $p-n-p$, для транзистора типа $n-p-n$ результаты имеют знаки, противоположные высвечиваемому (за исключением $g_{кб}$ и $g_{бб}$, знаки которых не зависят от типа транзистора).

Пример 71. Вычислить в рабочей точке слабосигнальные параметры транзистора и значения $U_{вб}$ и $U_{кб}$ для схемы, показанной на рис. 45, а, при $R_b = 1 \text{ кОм}$; $C_b = 0,02 \text{ мКФ} = 1/50 \text{ кОм}$; $R_k = 1 \text{ кОм}$; $G_1 = 0,01 \text{ мКФ} = 1/100 \text{ кОм}$; $G_2 = 0,05 \text{ мКФ} = 1/20 \text{ кОм}$; $R_6 = 0,5 \text{ кОм}$; $\alpha_N = 0,99$; $I_f = 0,01 \text{ мА}$; $E = -10 \text{ В}$.

По приведенным ранее формулам вычисляем $a = 1,01$; $b = 1,05$; $c = 1,025$; $d = 1,005$; $q = 2,59025$; $f = 1,117055$; $h = 1,08525$, а также коэффициенты A , B , C и D , вводим в УЗП программу 214 и в регистры памяти исходные данные $EA + 1 = 238,6921 = P2$; $B = -64,50354 = P3$; $CE = -3,943174 = P4$; $D = -0,5855535 = P5$; $\alpha_N = 0,99 = P6$; $I_f = 0,01 = P7$. По программе вычисляем $U_{вб} = 0,1394044 \text{ В}$; $U_{кб} = -4,024803 \text{ В}$; $I_s = 2,286997 \text{ мА}$; $g_{кб} = 88,30095 \text{ мКФ}$; $g_{бб} = 0,8919288 \text{ мКФ}$. В соответствии с приближенностью используемой модели полученные результаты следует округлить по крайней мере до трех значащих цифр.

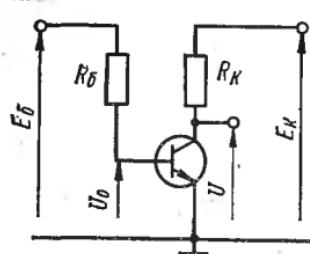


Рис. 46. Схема раздельного питания цепей базы и коллектора

В тех случаях, когда систему нелинейных уравнений, описывающих моделируемый объект, нельзя решить автоматически в связи с малой емкостью запоминающих устройств микрокалькулятора, можно попытаться найти решение переходом к дифференциальным уравнениям.

Для примера рассмотрим задачу вычисления $U_{кб}(E_b)$ при раздельном питании цепей базы и коллектора биполярного транзистора (рис. 46). Так как в этом случае допущение о достаточно большом значении напряжения

$U_{кэ}$ ограничивает область решения, опишем транзистор системой уравнений (149а), но для упрощения учтем сопротивление r'_b соответствующим увеличением сопротивления R_b . Обозначив $U_0 = U_{вб} - r'_b I_b$, опишем анализируемую схему системой уравнений

$$U_{кб} - A(e^{-\Lambda U_0} - 1) + B(e^{\Lambda(U_{кб}-U_0)} - 1) = E_b;$$

$$U_0 - C(e^{-\Lambda U_0} - 1) + D(e^{\Lambda(U_{кб}-U_0)} - 1) = E_b,$$

где $A = \alpha_N R_k I_{90}/M$; $B = R_k I_{k0}/M$; $C = (1 - \alpha_N) I_{90} R_b/M$;
 $D = (1 - \alpha_I) I_{k0} R_b/M$; $M = 1 - \alpha_N \alpha_I$.

Обозначив $\Delta E_b = x$, $\Delta U_{k0} = y$, $\Delta U_0 = z$ и продифференцировав уравнения по переменной x , получим систему дифференциальных уравнений

$$\begin{aligned} y'(1 + \Lambda B e^{y-z}) + z'(\Lambda A e^{-z} - \Lambda B e^{y-z}) &= 0; \\ y' \Lambda D e^{y-z} + z'(1 + \Lambda C e^{-z} - \Lambda D e^{y-z}) &= 1. \end{aligned}$$

Преобразуя эту систему относительно y' и z' , приведем ее к виду

$$\begin{aligned} y' &= (1 + \beta e^y)/\eta; \quad z' = (\delta e^z + \beta e^y)/\eta; \\ \eta &= \gamma + \delta e^z + \sigma e^y + \varepsilon e^{y-z}, \end{aligned}$$

где $\beta = -B/A = -I_{k0}/(\alpha_N I_{90})$; $\gamma = -C/A = - (1 - \alpha_N) I_{k0} R_b / (\alpha_N R_k I_{90})$;

$$\delta = -1/\Lambda A = -(1 - \alpha_N \alpha_I) / (\Lambda \alpha_N R_k I_{90});$$

$$\sigma = (D - B)/A = ((1 - \alpha_I) R_b - R_k) I_{k0} / (\alpha_N R_k I_{90});$$

$$\varepsilon = \Lambda (AD - BC)/A = \Lambda R_b I_{k0} (2\alpha_N - \alpha_N \alpha_I - 1) / \alpha_N (1 - \alpha_N \alpha_I).$$

Выбрав для решения метод Эйлера ($y_{i+1} = y_i + y'_i h$; $z_{i+1} = z_i + z'_i h$), составим программу 215 с исходными данными $\beta = P_2$, $\gamma = P_3$, $\delta = P_4$, $\sigma = P_5$, $\varepsilon = P_6$. В незаполненную часть программы записываем значение шага h по переменной $E_b = x/\Lambda$. После каждого выполнения программы (первый пуск по команде В/О, С/П) результаты оказываются занесенными в регистры $P7 = z_i$ и $PX = P8 = y_i$.

Программа 215. Вычисление зависимости напряжения на коллекторе от напряжения на базе биполярного транзистора

F8	e^x	\uparrow	F2	\times	\leftarrow	F5	\times	\leftarrow	F7	e^x	\div
\uparrow	F6	\times	\uparrow	F3	$+$	\leftarrow	F7	e^x	\uparrow	F4	\times
\uparrow	\rightarrow	$+$	XY	\rightarrow	$+$	$/-$	\dots	\dots	\div	$1/x$	
\rightarrow	1	$+$	\leftarrow	\leftarrow	$+$	\rightarrow	\uparrow	\rightarrow	\times	\leftarrow	\leftarrow
\times	\uparrow	F7	$+$	P7	\rightarrow	\rightarrow	\uparrow	F8	$+$	P8	C/P

Перед пуском программы 215 в регистры 7 и 8 необходимо записать начальные значения напряжений ΔU_0 и ΔU_{k0} , в качестве которых можно принять значения какого-либо тривиального решения. Например, при $E_b = 0$ в связи с малостью тока базы $U_0 \approx 0$, а для вычисления напряжения U_{k0} достаточно решить уравнение

$$U_{k00} + B(e^{\Delta U_{k00}} - 1) - E_k = 0,$$

которое при работе транзистора в типовом режиме с достаточной степенью точности дает значение $U_{k0} \approx E_k$.

Для транзисторов типа $p-n-p$ искомая зависимость представляется интерес при увеличении отрицательного напряжения на базе, что учтено в программе оператором $/—/$ по адресу 51.

Анализ нелинейных инерционных систем обычно проводят по математическим моделям в виде нелинейных дифференциальных уравнений. В этом случае возможность автоматизации решения задачи в значительной степени зависит от удачного преобразования и нормирования исходного уравнения и рационального выбора алгоритма вычислений.

В качестве примера рассмотрим высокочастотную детекторную цепь с полупроводниковым диодом, нагруженным на сопротивление R . С учетом нелинейных диффузионной и барьерной емкостей диода с несимметричным «резким» $p-n$ -переходом для напряжения $z(t)$ на диоде можно составить уравнение

$$x(t) = z(t) + R \left[I_0 (e^{\Lambda z(t)} - 1) + \left(\frac{\Lambda I_0 \tau}{2} e^{\Lambda z(t)} + \frac{\alpha}{V U_{bi} - z(t)} \right) \frac{dz(t)}{dt} \right]$$

или, относительно производной,

$$\frac{dz}{dt} = \frac{x - z + A (e^{\Lambda z} - 1)}{[\rho e^{\Lambda z} + 1/V U_{bi} - z] \gamma},$$

где $A = -I_0 R$; $\beta = \Lambda I_0 \tau / 2\alpha$; $\gamma = \alpha R$; $\Lambda = 39$.

Для решения этого уравнения при произвольном законе изменения $x(t)$ можно использовать программу 216 ($\theta/\gamma = P2$, $x_0 = P3$, $A = P5$, $\beta = P6$, $U_{bi} = P7$, $z = P8$, $x_i = PX$, $PX = z_i$).

Программа 216. Анализ переходных процессов в цепи с инерционным диодом

ПП	$P \times$	P3	XY	P8	ПП	$P \times$	+	2	\div	P8	C/P
БП	P0	P4	F8	3	9	\times	e^x	1	$-$	\rightarrow	F6
\div	\rightarrow	F7	\uparrow	F8	$-$	$\sqrt{ }$	$1/x$	\uparrow	\leftarrow	$+$	\leftarrow
\uparrow	F5	\times	\uparrow	F3	$+$	\uparrow	F8	\uparrow	\uparrow	\rightarrow	\div
\uparrow	F2	\times	\uparrow	F8	$+$	\uparrow	F4	B/O			

После каждого выполнения программы для введенного значения x_i высвечивается значение z_i напряжения на диоде, по которому несложно вычислить $y_i = x_i - z_i$.

В качестве другого примера рассмотрим анализ процессов, протекающих в резонансной цепи с отрицательным сопротивлением, реализуемым туннельным диодом при выборе рабочей точки на падающем участке его статической N-образной характеристики. Схема подобной цепи приведена на рис. 47, а.

Статическая характеристика туннельного диода достаточно точно аппроксимируется степенным многочленом пятой степени, однако при использовании такой модели не удается разместить программу вычислений и исходные данные в запоминающих устройствах микрокалькулятора с используемыми характеристиками. Поэтому ограничимся более приближенной моделью и предположим, что стати-

ческая характеристика активного двухполюсника, обведенного на рис. 47, *a* штриховой линией, аппроксимирована степенным многочленом

$$i(u) = a_0 + a_1 u + a_2 u^2 + a_3 u^3$$

с учетом выбора рабочей точки (определенной напряжением и внутренним сопротивлением источника питания) в средине падающего участка статической характеристики.

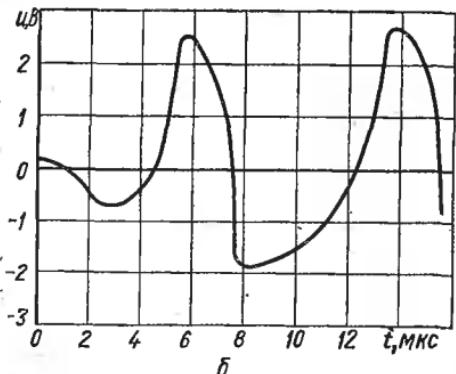
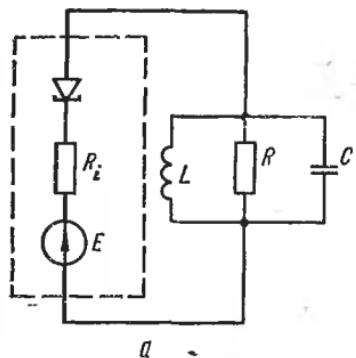


Рис. 47. Автогенератор на туннельном диоде:
a — эквивалентная схема; *б* — график установления колебаний

Дифференциальное уравнение рассматриваемой цепи, согласно первому закону Кирхгофа, имеет форму

$$i(u) + C \frac{du}{dt} + \frac{u}{r} + \frac{1}{L} \int u dt = 0$$

или, после дифференцирования,

$$\frac{d^2u}{dt^2} + \frac{1}{rC} \frac{du}{dt} + \frac{1}{C} \frac{di(u)}{du} \frac{du}{dt} + \omega_0^2 u = 0,$$

где $\omega_0 = 1/\sqrt{LC}$; $di(u)/du = a_1 + 2a_2 u + 3a_3 u^2$.

Вводя «безразмерное» время $x = \omega_0 t$ и обозначая

$$\varepsilon = \frac{1 + a_1 r}{\omega_0 r C}; \quad b = \frac{2a_2 r}{\sqrt{|3a_3 r(1 + a_1 r)|}}; \quad y = \sqrt{\left| \frac{3a_3 r}{1 + a_1 r} \right|} u,$$

приведем составленное уравнение к виду

$$\frac{d^2y}{dx^2} + \varepsilon (1 \pm by \pm y^2) \frac{dy}{dx} + y = 0,$$

Где знаки членов, стоящих в скобках, определяются с учетом знака подкоренного выражения.

Заменив $z = dy/dx$, представим уравнение в виде системы уравнений первого порядка

$$dz/dx = -\varepsilon (1 \pm by \pm y^2) z - y; \quad dy/dx = z.$$

Взяв за основу программу 141 решения системы из двух дифференциальных уравнений модифицированным методом Эйлера, дополним ее фрагментом вычисления производных dy/dx и dz/dx по значениям y и z , хранящимся в регистрах 5 и 7 соответственно, коэффициенту b , заносимому в регистр 4, и коэффициенту ϵ , записываемому непосредственно в текст программы по адресам 62—65. Шаг интегрирования и начальные условия записываются в регистры $h/2 = P2$, $x_0 = P3$, $y_0 = P5 = P6$, $z_0 = P7 = P8$. Полученная таким образом программа 217 вычисляет значения y и z , от которых несложно перейти к значениям $u(t)$ и $u'(t)$. Время одного выполнения программы составляет около 15 с.

Программа 217. Анализ автоколебаний в генераторе на туннельном диоде

ПП	F	÷	×	P2	F3	+	P3	ПП	F	÷	÷	P2	↑
F3	+			P3	F7	P8	F5	P6	C/П	БП	P0	F7	↑
F2	×		↑	F6	+	←	F5	↑	F4	—	×	1	
—	F7	×	↑	F5	—
↑	F2	×	↑	F8	+	P7	→	P5	F2	2	B/O		

В программе 217 при вычислении производной dz/dx приняты следующие знаки

$$dz/dx = -\epsilon(1 + by - y^2)z - y,$$

что при записи в тексте программы положительного значения ϵ соответствует неустойчивой системе. Если же анализируемая система устойчива и член y^2 , стоящий в скобках, положителен, операторы $|—|$ по адресам 54 и 61 следует заменить операторами $|+|$ и ввести оператор смены знака $/—/$ по адресу 62. В этом случае будет вычисляться выражение

$$dz/dx = -\epsilon(1 + by + y^2)z - y.$$

Работая с программой 217, необходимо учитывать, что при значениях коэффициента $|\epsilon| > 3$ для обеспечения сходимости шаг интегрирования приходится выбирать очень малым, а это существенно увеличивает время анализа. Поэтому при больших значениях ϵ может оказаться целесообразным решать составленную систему уравнений первого порядка классическим методом Эйлера, позволяющим получить устойчивое решение при большем шаге интегрирования, хотя и с меньшей точностью.

Пример 72. Построить зависимость $u(t)$ при включении источника питания в цепь, эквивалентная схема которой показана на рис. 47, а, при $r = 0,25$ кОм, $L = 50$ мГн, $C = 800$ пФ, $i = 30 - 10u - 2u^2 + 0,8u^3$ (ток в мА, напряжение в В).

По исходным данным вычисляем и записываем в программу со своим знаком значение коэффициента $\epsilon = -1,5$, а в регистры памяти заносим принятые или вычисленные исходные данные $h/2 = \pi/40 = P2$, $b = 1,054092 = P4$, $x_0 = z_0 = 0 = P3 = P7 = P8$, $y_0 = 0,1 = P5 = P6$. Результаты вычислений, характеризующие процесс установления автоколебаний в анализируемой цепи, изображены в виде графика $y(t)$ на рис. 47, б, где значения $y(t)$ пропорциональны значениям $u(t)$.

Если арифметическое выражение для решения дифференциального уравнения не удается уместить в УЗП, следует попытаться найти решение задачи с помощью цифрового моделирования, используя простейшие аппроксимации характеристик нелинейных звеньев.

Для примера рассмотрим задачу моделирования процессов в физической системе, моделируемой соединением двухстороннего ограничителя и резонатора (рис. 48, *a*). К подобной модели сводится, в частности, резонансный усилитель электрических сигналов при большом уровне последних. Передача двухстороннего ограничителя

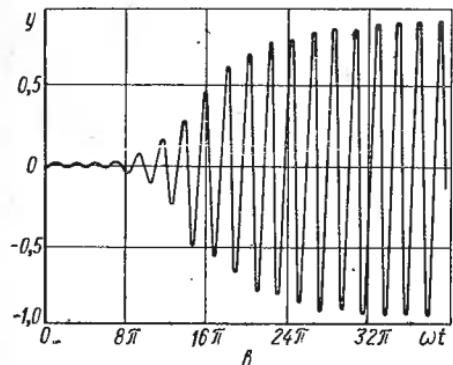
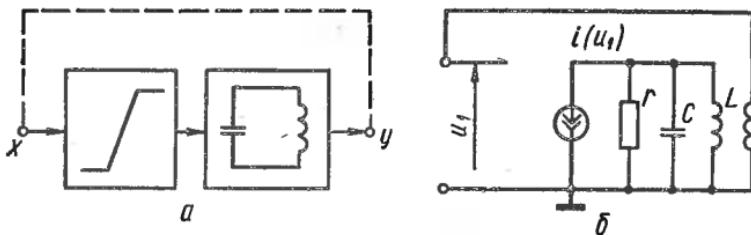


Рис. 48. Канал с двусторонним ограничителем и резонатором:
а — обобщенная схема;
б — электрическая схема;
в — график установления автоколебаний

в рабочей области (см. рис. 34, *б*) в этом случае равна средней крутизне S характеристики прямой передачи безынерционного усилителя, а уровни ограничения определяются свойствами активного элемента.

Объединяя программы 78 и 88 для вычисления передач звеньев рассматриваемой системы, получаем программу 218 ($W_0 = P2$, $W_1/W_2 = P3$, $-W_2 = P4$, $S = P5$, $y_u = P6$, $y_b = P7$, $x_i = PX$, $PX = y_t$).

Программа 218. Цифровая модель соединения двухстороннего ограничителя и резонатора

\uparrow	F5	\times	\uparrow	F7	P8	—	$x < 0$	F	\times	F6	P8	—
$x \geqslant 0$	F	\times	XY	P8	F8	\uparrow	\leftarrow	\leftarrow	—	\leftarrow	\uparrow	F3
\times	XY	\leftarrow	+	\uparrow	F4	\times	\uparrow	\rightarrow	\rightarrow	\rightarrow	+	\uparrow

Если перед первым пуском программы 218 ввести в регистр X малое число, имитирующее естественные флюктуации, а в дальнейшем не изменять содержание регистра X, нажимая только кла-

вишу С/П, то результаты последующих вычислений будут отображать поведение моделируемой системы при ее охвате положительной (так как в регистре Х в качестве переменной воздействия сохраняется предыдущее значение реакции с тем же знаком) обратной связью. Если при этом соблюдаются условия самовозбуждения, то с помощью программы 218 можно анализировать возникновение и установление автоколебаний в моделируемой системе.

Пример 73. Исследовать установление автоколебаний в цепи, образованной резонансным усилителем с положительной обратной связью (рис. 48, б). Характеристика прямой передачи безынерционного усилительного элемента описана параметрами двустороннего ограничителя $y_u = -10\text{ mA}$, $y_b = 10\text{ mA}$. $S = 30\text{ mA/B}$; входная и выходная проводимости усилительного элемента учтены в параметрах колебательного контура $C = 10^3/2\pi\text{ нФ}$, $L = 10/2\pi\text{ мГн}$, $R = 1\text{ кОм}$, а знак и глубина обратной связи учтены в значении параметра S .

Определив резонансную частоту колебательного контура $f_p = 1/\pi\sqrt{LC} = 10\text{ кГц}$, выбираем интервал дискретизации $\theta = 10\text{ мкс}$. В соответствии с формулой (144) находим $\omega_A = \operatorname{tg}\omega_p\theta/2 = 0,3249196$ и вычисляем коэффициенты $W_1 = 35,02604$; $W_2 = -55,05529$; $W_3 = 33,02603$. После занесения в УЗП программы 218 вводим исходные данные $35,02604 = P2$; $-1,667026 = P3$; $-33,02603 = P4$; $30 = P5$; $-10 = P6$; $10 = P7$; $0,01 = P8$. Результаты вычисленных по программе 218 при этих исходных данных изображены в виде графика на рис. 48, б.

Изменяя значение $S = P5$ перед каждым пуском программы 218, можно изменять (модулировать) амплитуду автоколебаний, причем при уменьшении значения крутизны S до некоторого критического предела условия самовозбуждения перестают выполняться и автоколебания прекращаются.

Если перед каждым пуском программы 218 в индикаторный регистр вводить очередное значение входного воздействия, обратная связь будет разорвана (так как значения реакции в регистре Х стираются) и программа будет представлять собой цифровую модель нелинейного резонансного усилителя, исходную рабочую точку которого можно определять как параметрами двустороннего ограничения, так и значением постоянной составляющей входного воздействия. Это позволяет моделировать прохождение сигнала через такие устройства как линейный резонансный усилитель (при малом воздействии и надлежащем выборе рабочей точки), нелинейный усилитель — ограничитель, модулируемый генератор с внешним возбуждением, умножитель частоты и т. п.

Подобные цифровые модели можно составлять и для нелинейных инерционных каналов других типов. Рассмотрим, например, логарифмический усилитель на интегральной микросхеме с коэффициентом передачи напряжения $K = K_0 \ln(1 + |U_1|)$ при инерционных элементах связи (рис. 49, а).

Коэффициент передачи входной цепи усилителя

$$K_1(p) = U_1(p)/U_{\text{вх}}(p) = p\tau_1/(1 + p\tau_1),$$

где $\tau_1 = R_1 C_1$ после z -преобразования

$$K_1(z) = (1 - z^{-1})\tau_1/[1 + \tau_1 + (1 - \tau_1)z^{-1}]$$

описывается разностным уравнением

$$U_1(i) = \frac{[U_{\text{вх}}(i) - U_{\text{вх}}(i-1) + U_1(i-1)] \tau_1 - U_1(i-1)}{1 + \tau_1}$$

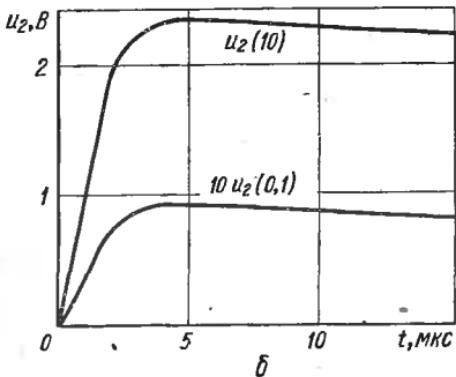
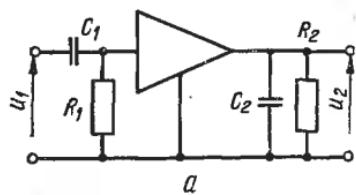
Коэффициент передачи выходной цепи, связывающий выходное напряжение с реакцией на выходе усилителя,

$$K_3(p) = U_{\text{вых}}(p)/U_2(p) = 1/(1 + p\tau_2),$$

где $\tau_2 = R_2 C_2$, после z -преобразования

$$K_3(z) = (1 + z^{-1})/[1 + \tau_2 + (1 - \tau_2) z^{-1}]$$

Рис. 49. Логарифмический усилитель:
а — схема; б — переходная характеристика



описывается разностным уравнением

$$U_{\text{вых}}(i) = \frac{U_3(i) + U_3(i-1) - (1 - \tau_2) U_{\text{вых}}(i-1)}{1 + \tau_2}.$$

Объединяя цифровые модели собственно усилителя, входной и выходной цепей, составим программу 219 с исходными данными $\tau_1 = P4$, $\tau_2 = P7$, $K_0 = P8$, $U_{\text{вх}} = RX$. Значение реакции $U_{\text{вых}}(i)$ выводится на индикатор. Для первого пуска программы нажимают клавиши В/О и С/П, для последующих — только клавишу С/П. Время каждого выполнения программы около 13 с.

Программа 219. Цифровая модель логарифмического усилителя с инерционными элементами связи

\uparrow	F2	\rightarrow	XY	P2	F3	$+$	\uparrow	F4	\times	\uparrow	F3
\leftarrow	F4	1	$+$	\uparrow		\rightarrow	XY	\div	P3	$x < 0$	F4
$/-$	1	$+$	ln	\uparrow	F8	\times	\rightarrow	F7	1	$-$	\uparrow
F6	\times	\uparrow	F5	$+$	\uparrow	\rightarrow	P5	$+$	\rightarrow	F7	1
$+$	\uparrow	\leftarrow	XY	\div	P6	C/P	BП	P0			

Пример 74. Определить реакцию $u_{\text{вых}}(t)$ на ступенчатое воздействие $u_{\text{вх}}(t) = 0$ при $t < 0$, $u_{\text{вх}}(t) = U$ при $t \geq 0$ логарифмического усилителя (рис. 49, а) при $R_1 = 10 \text{ кОм}$, $R_2 = 1 \text{ кОм}$, $C_1 = 10 \text{ нФ}$, $C_2 = 1 \text{ нФ}$, $K_0 = 1$.

По значениям $\tau_1 = 10^{-4} \text{ с}$ и $\tau_2 = 10^{-6} \text{ с}$ определим частоты $\omega_{\text{ц1}}$ и $\omega_{\text{ц2}}$, соответствующие границам полосы пропускания входной и выходной цепей усилителя.

теля $\omega_{\text{д}1} = 10^4$ и $\omega_{\text{д}2} = 10^6$. Выбрав шаг дискретизации $\theta = 1$ мкс в соответствии с формулой (144) находим $\omega_{A1} = 0,005$; $\omega_{A2} = 0,5463026$, по которым определим постоянные времени моделируемых цепей $\tau_{A1} = 1/\omega_{A1} = 200$ и $\tau_{A2} = 1/\omega_{A2} = 1,830487$.

Результаты вычислений для двух значений $U = 0,1$ и 10В ступенчатого входного воздействия изображены в виде графиков на рис. 49, б, причем для удобства сравнения масштаб реакции на воздействие меньшего уровня увеличен в 20 раз.

В рассмотренных выше цифровых моделях пренебрегалось зависимостью свойств безынерционного нелинейного звена от значе-

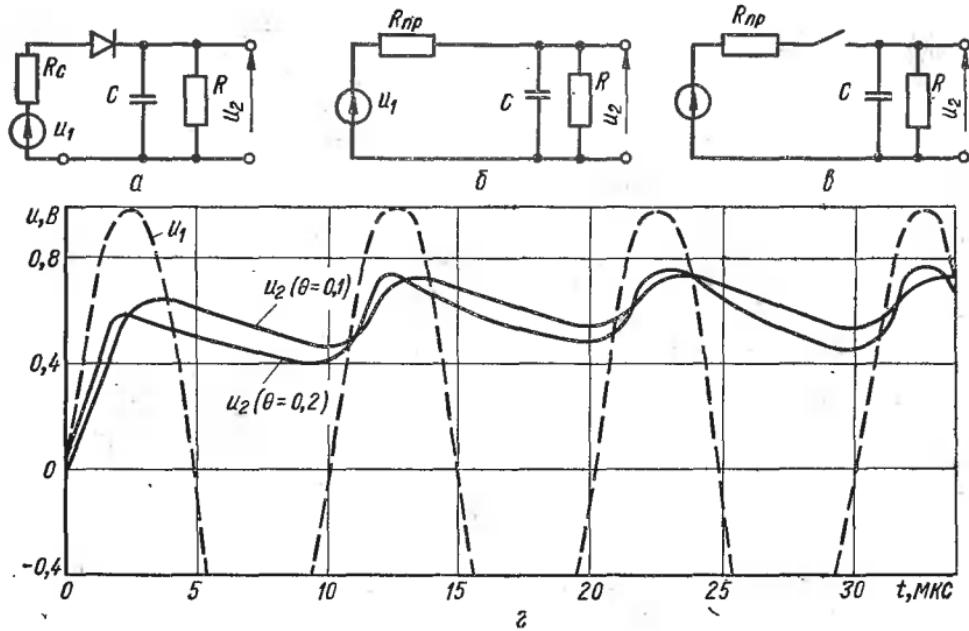


Рис. 50. Диодный детектор:

а, б, в — схемы замещения; г — графики выходного напряжения

ния переменной реакции. Для некоторых физических объектов подобное предположение недопустимо, так как вносит принципиальную погрешность в модель. Простейшим примером может служить диодный детектор (рис. 50, а), используемый для выделения огибающей (детектирования) высокочастотного электрического сигнала. В подобном детекторе выходная цепь накапливает электрическую энергию и напряжение на диоде (следовательно, и ток через диод) зависит как от входного, так и от выходного напряжения, равняясь их разности.

При относительно большом уровне входного сигнала («линейное» детектирование) статическую характеристику диода с достаточной точностью можно аппроксимировать кусочно-линейной характеристикой идеального вентиля, не пропускающего ток в закрытом состоянии и имеющего сопротивление прямой передачи $R_{\text{пр}}$,

включающее и внутреннее сопротивление источника воздействия, в открытом состоянии. Таким образом, при подобной аппроксимации диод можно рассматривать как коммутирующий ключ, замыкающийся при $u_1 > u_2$ (рис. 50, б) и размыкающийся при $u_1 < u_2$ (рис. 50, в). В каждом из этих состояний связь реакции с воздействием описывается линейным операторным уравнением

$$U_2(p) = \begin{cases} U_1(p)/[1 + (1 + p\tau) R_{\text{пр}}/R] & \text{при } u_1 > u_2; \\ p\tau U_2(0)/(1 + p\tau) & \text{при } u_1 < u_2, \end{cases}$$

где $\tau = RC$ и $U_2(0)$ — напряжение на емкости детектора к моменту запирания диода. Используя z -преобразование, составим соответствующие разностные уравнения

$$U_2(i) = \begin{cases} [u_1(i) + u_1(i-1) - Bu_2(i-1)]/A & \text{при } u_1 > u_2; \\ -Gu_2(i-1) & \text{при } u_1 < u_2, \end{cases}$$

где $A = 1 + (1 + \tau) R_{\text{пр}}/R$; $B = 1 + (1 - \tau) R_{\text{пр}}/R$; $G = (1 - \tau)/(1 + \tau)$.

Следовательно, программа для вычисления реакции детектора по этим формулам должна содержать фрагмент, сравнивающий напряжения $u_1(i)$ и $u_2(i-1)$ и в зависимости от их соотношения выбирающий расчетную формулу. Подобный алгоритм реализован программой 220 ($A = P4$, $-B = P5$, $-G = P6$, $u_1(i) = PX$, $PX = u_2(i)$).

Программа 220. Цифровая модель диодного детектора при большом значении $\tau/(R_{\text{пр}})$

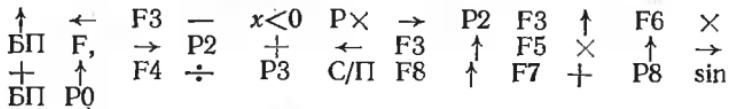
\uparrow БП $+$ F3	\leftarrow $/-$ \uparrow F4	F3 БП	$-$ P2 \div P0	$x < 0$ $+$ \uparrow P3	$P \times$ \leftarrow \uparrow F2	P2 F3 \uparrow F5	\rightarrow \uparrow \uparrow x $\geqslant 0$	F6 \times \uparrow P6	\times \uparrow \uparrow F2	\times \rightarrow P3
-------------------------------	--	----------	---------------------------	------------------------------------	--	------------------------------	--	------------------------------------	--	---------------------------------

В процессе вычислений значения $u_1(i)$ и $u_2(i)$ заносятся соответственно в регистры 2 и 3 для использования при следующем выполнении программы в качестве значений $u_1(i-1)$ и $u_2(i-1)$. При большой постоянной времени нагрузки детектора и малом отношении $R_{\text{пр}}/R$ выбор чрезмерно большого шага дискретизации может привести к заметной методической погрешности, проявляющейся в том, что при возрастании входного воздействия максимальное значение реакции может оказаться большим максимального значения воздействия, что физически невозможно. Для устранения подобной погрешности в программу 220 введен специальный фрагмент, содержащий операторы с адресами от 51 до 65 включительно. При воздействии достаточно простой формы цифровую модель 220 можно дополнить фрагментом для автоматического формирования следующего значения входного воздействия.

Пример 75. Определить реакцию диодного детектора (рис. 50, а) с $R_1 = 10\text{k}\Omega$, $R_{\text{пр}} = 1 \text{ к}\Omega$, $C = 2000 \text{ пФ}$ на включение синусоидального воздействия $u_1 = u_m \sin 2\pi \cdot 10^6 t$.

В соответствии с заданными параметрами детектора при выборе достаточно малого значения θ можно упростить цифровую модель 220, изъяв из нее операторы с адресами 51—65. Вместе с тем для ускорения вычислений целесообразно дополнить цифровую модель детектора при заданном воздействии генератором табличной модели синусоиды. В результате получим программу 221 с исходными данными $A = \text{P}4$, $-B = \text{P}5$, $-G = \text{P}6$, $2\pi\theta/T = \text{P}7$, $0 = \text{P}8 = \text{P}X$.

Программа 221. Цифровая модель диодного детектора при синусоидальном воздействии



Приняв $\theta = 0,1 \text{ мкс}$, находим $\omega_A = \operatorname{tg}(\omega_u \theta / 2) = 0,3249196$. Тогда $\tau_A = (\omega_u \tau) / \omega_A = 38,67532$ и $A = 4,967532$, $B = -2,767532$, $G = -0,9495908$, $2\pi f\theta = \pi/5$. Занеся программу 221 и исходные данные в запоминающие устройства микрокалькулятора, вычисляем отсчеты реакции детектора.

Для оценки точности полученных результатов примем $\theta = 0,2 \text{ мкс}$ и повторим проведенные вычисления (в этом случае $\omega_A = 0,7265425$, $\tau_A = 17,29612$, $A = 2,829612$, $B = -0,629612$, $G = -0,8906872$, $2\pi f\theta = 2\pi/5$). Из сравнения полученных результатов (рис. 50, г) видно, что достигнутая точность достаточна для инженерных расчетов.

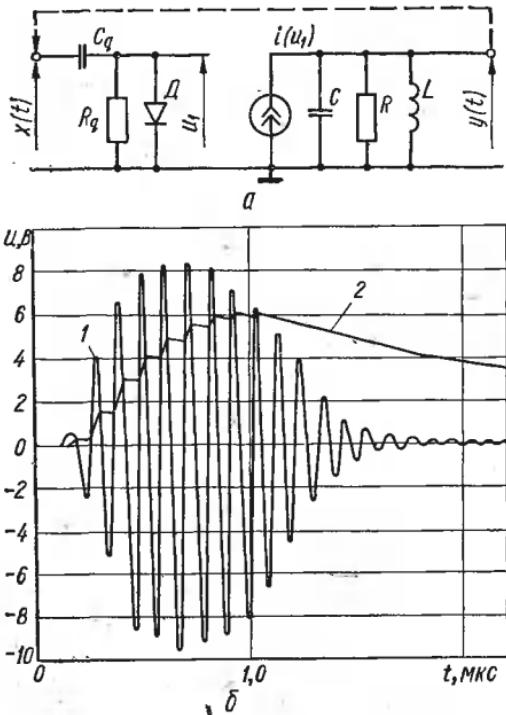
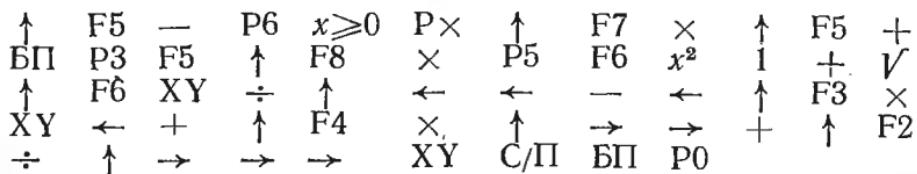


Рис. 51. Цель с прерывистой генерацией:
а — эквивалентная схема; б — график выходных напряжений

моделью подобной системы является процессы в цепи, эквивалентная рис. 51, а.

Иногда явление детектирования, возникающее в нелинейных системах и проявляющееся в относительно медленном изменении режима питания активных компонентов, оказывает существенное влияние на поведение системы в целом. Примером может служить прерывистая генерация, возникающая вследствие детектирования генерируемых колебаний нелинейностью входной цепи активного компонента. Алгоритмической программой 222, описывающей схему которой показана на

**Программа 222. Модель автоколебательной системы
с прерывистой генерацией**



При $Q = 5$, $f_p = 10 \text{ МГц}$, $R = 1 \text{ кОм}$, $S_0 = 10 \text{ мА/В}$, постоянных времени цепи заряда $\tau_{\text{зар}} = 0,1 \text{ мкс}$ и разряда $\tau_{\text{пaa}} = 2 \text{ мкс}$ в рассматриваемой цепи возникает прерывистая генерация. В программе использована аппроксимация характеристики входной цепи отрезками прямых и тока зависимого источника выражением

$$i(u_1) = S_0 u_1 / \sqrt{1 + u_1^2}.$$

По программе вычисляется $i(u_1) = u_1 / \sqrt{1 + u_1^2}$, поэтому значение S_0 необходимо учесть при определении коэффициентов цифрового резонатора, которые определяются как

$$W_0 = \frac{(1 + \omega_p^2 + \omega_p d)}{\omega_p d R S_0}; \quad W_1 = \frac{2(\omega_p^2 - 1)}{\omega_p d R S_0}; \quad W_2 = \frac{1 + \omega_p^2 - \omega_p d}{\omega_p d R S_0}.$$

Исходные данные вводятся в регистры $W_0 = P2$, $W_1/W_2 = P3$, $-W_2 = P4$, $1 - e^{-\theta/\tau_{\text{зар}}} \approx \theta/\tau_{\text{зар}} = P7$, $e^{-\theta/\tau_{\text{пaa}}} \approx 1 - \theta/\tau_{\text{пaa}} = P8$.

Результаты вычислений по программе 222 для $\theta = 10 \text{ нс}$ приведенных выше исходных данных и начального значения $x(0) = 0,01$ изображены в виде графика на рис. 51, б, показанного сплошной линией. Штриховой линией на том же рисунке показано напряжение на конденсаторе C_q , заносимое в процессе выполнения программы в регистр 5.

Рассмотренные примеры, естественно, не могут охватить все многообразие даже сравнительно простых задач нелинейного моделирования и возникающих при их решении трудностей, но характеризуют ряд приемов, позволяющих автоматизировать решение таких задач с помощью программируемых микрокалькуляторов, имеющих ограниченную емкость запоминающих устройств.

ПРИЛОЖЕНИЕ

Приведенные в настоящей книге программы решения типовых задач, в основном составленные на входном языке микрокалькулятора типа «Электроника Б3-21», можно перевести на входной язык другого настольного или карманного программируемого или непрограммируемого калькулятора. В не-программируемом калькуляторе достаточно перевести неразветвленные части программы и, если они входят в итерационные циклы, составить вычислительный блок для записи результатов вычислений на каждой итерации.

При переводе программ, как и при их непосредственном использовании, следует учитывать, что серийные микрокалькуляторы непрерывно совершенствуются и их отдельные выпуски могут отличаться определенными особенностями. В частности, отмеченные на стр. 192 недостатки ранних образцов микрокалькулятора типа «Электроника Б3-21» устранены в последующих выпусках. При этом возникла следующая особенность — результат вычисления разности двух чисел, из которых первое больше, а второе меньше 10^{-8} , равен первому числу, уменьшенному на единицу последнего разряда мантиссы (например, $10 - 10^{-22} = 9,9999999$ или $5,5555555 - 10^{-67} = 5,5555554$).

Эта особенность микрокалькуляторов поздних выпусков практически не влияет на точность результатов вычислений, но при реализации приведенным на стр. 209 фрагментом « $XY + P7 -$ » критерия (140) для суммирования знакопостоянных рядов время вычислений резко возрастает. Подобный недостаток можно устранить, использовав фрагмент с таким же числом операторов (чтобы не изменять адресации переходов в программах), но не содержащий оператора вычитания. В качестве такого фрагмента, например в программах 42, 110, 111 и 113, можно использовать « $+P7 \div \sin$ », что обеспечит их пригодность для вычислений на микрокалькуляторах любого выпуска. Однако, если такой фрагмент расположить в программе между операторами поворота стека, то при использовании микрокалькуляторов ранних выпусков может возникнуть неконтролируемое переполнение регистров стека (см. стр. 192).

Если, например, ввести подобный фрагмент в программу 121, то она станет пригодной для микрокалькуляторов всех выпусков, но при последующем использовании программы 124 на микрокалькуляторе раннего выпуска может возникнуть переполнение в стеке памяти. Поэтому для пригодности программы 121 как промежуточной при вычислениях корней многочленов высшего порядка на микрокалькуляторах всех выпусков ее следует использовать в приводимой здесь форме.

Программа 121. Вычисление вещественных корней кубического многочлена

F4	P6	F6	2	\times	P6	$/-$	ПП	6	$x < 0$	Р↑	F6
2	\div	P6	\uparrow	F5	$+$	P5	XY	x^2	\checkmark	$+$	$-$
$x = 0$	F;	F5	C/P	ПП	FВП	$x \geq 0$	XY	F5	\uparrow	F6	$-$
P5	БП	XY	P5	F5	\uparrow	F2	ПП	$-$	F3	ПП	$-$
F4	XY	$+$	\times	B/O	\uparrow	\uparrow	F5	\times	\uparrow	B/O	

Аналогично программа 129 окажется пригодной для вычислений на микрокалькуляторах всех выпусков при ее преобразовании к приводимой здесь форме.

Программа 129. Вычисление вещественного корня нормированного уравнения степени 7

F6	2	\div	P6	\uparrow	F7	$+$	P7	XY	x^3	\checkmark	$+$
$-$	$x = 0$	\times	F7	C/P	F7	\uparrow	F2	$+$	ПП	C _x	ПП
C _x	ПП	C _x	ПП	C _x	ПП	C _x	ПП	C _x	\uparrow	F3	\leftarrow
F8	XY	P8	\times	$x < 0$	P0	F6	$/-$	P6	БП	P0	\uparrow
F7	\times	\uparrow	F3	\leftarrow	P3	$+$	B/O				

Перевод операторов перехода и передачи управления с простого входного языка микрокалькулятора типа «Электроника Б3-21» на входные языки программируемых микрокалькуляторов других типов не вызывает трудностей, так как подобные операторы имеются во всех таких языках, а передача управления реализуется в них более простыми способами.

Основные трудности возникают при переводе фрагментов программ с двухместными операторами на входные языки программируемых или непрограммируемых микрокалькуляторов с синтаксическими различиями, связанными с конструкцией и емкостью операционных устройств. Это особенно существенно, когда в заданном арифметическом выражении имеются старшие operandы, которые должны быть вычислены до выполнения операции, например, в выражении

$$((a \times b) - (c \times d)) : ((e \times f) - (g : h)).$$

Подобные выражения с operandами второго порядка старшинства (имеющие скобки в скобках) можно описать формулой

$$((aO_1b)O_2(cO_3d))O_4((eO_5f)O_6(gO_7h)),$$

где O_i — любой двухместный оператор. Результаты операций деления и вычитания зависят от порядка ввода operandов. Поэтому программа для вычисления подобного арифметического выражения на микрокалькуляторе типа «Электроника Б3-21» (с двухрегистровым операционным устройством и обратным синтаксисом входного языка) будет иметь следующую структуру:

$$\begin{array}{ccccccccc} a & \uparrow & b & O_1 & P_1 & c & \uparrow & d & O_3 \\ F_2 & O_2 & P_3 & e & \uparrow & f & O_5 & P_4 & g \\ P_5 & F_4 & \uparrow & F_5 & O_6 & P_6 & F_1 & F_6 & O_4 \end{array} \quad \begin{array}{c} P_2 \\ F_1 \\ \uparrow \\ O_7 \end{array}$$

Здесь цифрами N в операторах PN и FN указаны не номера регистров памяти, а порядковые номера результатов промежуточных вычислений, регистрируемых в памяти или вызываемые из нее.

Перевод подобной программы на входной язык микрокалькулятора с обратным синтаксисом, но с трехрегисторным операционным устройством (например, микрокалькулятора типа «Электроника Б3-19»), будет иметь меньшее число операторов и лишь два обращения к памяти:

$$\begin{array}{ccccccccc} a & \uparrow & b & O_1 & \uparrow & c & \uparrow & d & O_3 \\ \uparrow & f & O_5 & \uparrow & g & \uparrow & h & O_7 & O_6 \\ F_2 & O_4 & & & & & & P_2 & F_1 \end{array} \quad \begin{array}{c} e \\ \uparrow \\ O_2 \\ P_1 \end{array}$$

При использовании микрокалькулятора с обратным синтаксисом входного языка, но с четырехрегисторным операционным стеком (например, микрокалькулятора типа «Электроника Б3-34»), перевод исходной программы будет содержать минимальное число операторов и не требует обращения к памяти:

$$\begin{array}{ccccccccc} a & \uparrow & b & O_1 & \uparrow & c & \uparrow & d & O_3 \\ \uparrow & f & O_5 & \uparrow & g & \uparrow & h & O_7 & O_6 \\ & & & & & & & O_4 & \uparrow \end{array} \quad \begin{array}{c} e \\ \uparrow \\ O_2 \\ P_1 \end{array}$$

Перевод исходной программы на входной язык с алгебраическим синтаксисом при операционном устройстве для одновременного хранения только двух operandов (как в большинстве арифметических микрокалькуляторов) будет иметь следующую структуру:

$$\begin{array}{ccccccccc} a & O_1 & b & = & P_1 & c & O_3 & d & = P_2 \\ F_2 & = & P_3 & e & O_5 & f & = & P_4 & d \\ P_5 & F_4 & O_6 & F_5 & = & P_6 & F_3 & O_4 & F_6 = \end{array}$$

Следовательно, этот перевод при том же числе обращений к памяти по числу операторов примерно соответствует исходной программе.

Перевод на входной язык микрокалькулятора с алгебраическим синтаксисом входного языка, но с операционным устройством для хранения трех operandов (что соответствует наличию во входном языке оператора простых скобок), будет иметь более простую структуру:

$$\begin{array}{c} a \quad O_1 \quad b \quad O_2 \quad (\quad c \quad O_3 \quad d \quad) = P1 \quad e \\ O_5 \quad f \quad O_6 \quad (\quad g \quad O_7 \quad h \quad) = P2 \quad F1 \quad O_4 \\ F2 = \end{array}$$

Если микрокалькулятор, на входной язык которого выполняется перевод, имеет операторы двойных скобок (в операционном устройстве одновременно хранится до четырех операндов), как, например, микрокалькулятор типа «Электроника Б3-36», то перевод имеет наименее простую структуру:

$$\begin{array}{c} a \quad O_1 \quad b \quad O_2 \quad (\quad c \quad O_3 \quad d \quad) \quad O_4 \quad (\\ e \quad O_5 \quad f \quad O_6 \quad (\quad g \quad O_7 \quad h \quad) \quad) = . \end{array}$$

При переводе сложных арифметических выражений часто оказывается целесообразным восстановить по программе-оригиналу алгоритм вычислений, откорректировать этот алгоритм в соответствии с особенностями входного языка, на который переводится программа, а затем составить программу-перевод по откорректированному алгоритму.

Так как двухместные арифметические операторы содержатся во входных языках всех микрокалькуляторов, то их словарный запас может отличаться только одноместными и немногочисленными двухместными операторами вычисления функций (к последним во входном языке микрокалькулятора типа «Электроника Б3-21» относится только оператор X^y). При переводе одноместных операторов синтаксические затруднения не возникают, так как практически во всех входных языках используется обратная запись оператора после операнда, но, как и в обычных языках, трудности возникают при лексических различиях в языках. Перевод одноместных операторов на входные языки, в которых они отсутствуют, приходится выполнять с помощью арифметических выражений, содержащих символы операций, автоматическое выполнение которых предусмотрено в языке перевода. Для этого можно использовать, в частности, приводимые в первых главах книги выражения для вычисления функций при помощи только арифметических операторов. Перевод таких функций, как $y = 1/x$ или $y = x^n$, обычно удается реализовать с помощью особенностей операционного устройства микрокалькулятора, для которого выполняется перевод.

Специфичным во входном языке микрокалькулятора типа «Электроника Б3-21» является оператор e^{ix} , который в зависимости от его назначения в программе переводится с помощью различных операторов вычисления тригонометрических функций или, при их отсутствии во входном языке перевода, арифметическими выражениями с арифметическими операторами.

При переводе программ существенно полное использование всех возможностей микрокалькулятора, связанных с его устройствами памяти и, особенно, операционным устройством, для составления перевода с минимальным количеством операторов и минимальным числом обращений к внешней памяти (функции которой выполняет вычислительный блок). Следует помнить, что каждое обращение к внешней памяти увеличивает время вычислений как непосредственно, так и в связи с увеличением вероятности ошибок при регистрации и считывании промежуточных результатов на вычислительном бланке и затрат времени на устранение этих ошибок.

Приведенными в книге программами могут воспользоваться обладатели не только отечественных, но и зарубежных непрограммируемых и программируемых микрокалькуляторов. Из последних практически любой можно отнести к одной из двух основных групп, отличающихся синтаксическими особенностями записи двухместных операций и способами передачи управления при условных и безусловных переходах.

Типичными представителями первой группы являются программируемые микрокалькуляторы фирмы Хюллет—Паккард (например, HP-25, HP-33, HP-41C, HP-67, HP-97). Входной язык этих микрокалькуляторов, как и микрокалькуляторов типа «Электроника Б3-21» или Электроника Б3-34», отличается обратным синтаксисом одноместных и двухместных операций. Смещение «вверх» операционного стека реализуется оператором «ENTER↑»

(enter — ввести), но при наборе числа после любой операции и при вызове в индикаторный регистр содержимого регистра памяти операционный стек, как и в микрокалькуляторе типа «Электроника Б3-34», смещается «вверх» автоматически.

Оператор безусловного перехода в зарубежных микрокалькуляторах обычно обозначен символом «GTO» (go to — идти на), после которого вводят две цифры, указывающие порядковый номер оператора (шага) программы, которому передается управление.

После операторов условного перехода в зарубежных микрокалькуляторах первой группы вводят оператор «GTO» и порядковый двузначный номер оператора программы, которому передается управление по адресу «Да». Управление по адресу «Нет» (в отличие от отечественных программируемых микрокалькуляторов) передается оператору, следующему в программе за оператором передачи управления по адресу «Да».

Первый шаг программы (с адресом 00) отводится для записи кода оператора «PRGM» (программа). В конце программы записывается оператор «RTN» (return — возврат), поворачивающий стек программной памяти в начальное положение, или оператор «GTO A», где А — номер шага программы (в частности, начального), которому передается управление. В некоторых микрокалькуляторах выполнение программы автоматически прекращается после выполнения последнего оператора, и в этих случаях операторы перехода в конце программы можно не вводить.

В зарубежных микрокалькуляторах второй группы, типичными представителями которых являются микрокалькуляторы фирмы Texas Instruments (например, TI P57 или TI P58), входной язык отличается обычной алгебраической формой записи двухместных операций и передачей управления при переходе с помощью оператора метки «LBL» (label — метка). В этом случае после оператора безусловного или условного перехода вводится оператор метки и цифровой или буквенный индекс метки. Точно такой же оператор (метки и индекса метки) вводится в программу перед оператором, которому передается управление.

Обмен информацией с регистрами памяти практически во всех зарубежных микрокалькуляторах реализуется с помощью операторов «STO» (storage—запасать) и «RCL» (recall — вызывать), после которых указывается номер соответствующего регистра памяти.

Оператор перехода к подпрограмме обычно обозначается символами «GSB» (go to subroutine — перейти к подпрограмме) или «SBR» (subroutine — подпрограмма). В конце подпрограммы записывается оператор «RTN» для возвращения к переходу к подпрограмме.

При редактировании программы в зарубежных микрокалькуляторах используют операторы «SST» и «BST», аналогичные операторам $\overleftarrow{ШГ}$ и $\overleftarrow{ШГ}$ микрокалькулятора типа «Электроника Б3-21», а также оператор «NOP» (нет операции) для стирания ошибочно введенного оператора.

Оператор ввода порядка обычно обозначают символами «EEX» или «EE» (enter exponent — ввод порядка), а оператор смены знака — как «CHS» (change sign — изменение знака или $+/-$). Из других встречающихся во входных языках программируемых зарубежных микрокалькуляторов операторов можно упомянуть оператор «PAUSE» (пауза) для временной (на несколько секунд) остановки программы и высвечивания результата промежуточных вычислений (например, при использовании итерационных алгоритмов) для его регистрации или контроля.

СПИСОК ЛИТЕРАТУРЫ

1. Айвазян С. А. Статистические исследования зависимостей. М., Металлургия, 1968.
2. Анго А. Математика для электро- и радиоинженеров. М., Наука, 1964.
3. Бронштейн И. Я., Семеняев К. А. Справочник по математике. М., Физматгиз, 1962.
4. Варакин А. Е. Теория сложных сигналов. М., Сов. радио, 1970.
5. Гарднер М. Математические головоломки и развлечения. М., Мир, 1971.
6. Голд Б., Рэйдер Ч. Цифровая обработка сигналов. М., Сов. радио, 1973.
7. Гутер Р. С., Овчинский Б. В. Элементы численного анализа и математической обработки результатов эксперимента. М., Наука, 1970.
8. Демидович Б. П., Марон И. А. Основы вычислительной математики. М., Наука, 1970.
9. Диткин В. А., Прудников А. П. Интегральные преобразования и операционное исчисление. М., Физматгиз, 1961.
10. Знаменский А. Е., Теплюк И. Н. Активные RC-фильтры. М., Связь, 1970.
11. Зуховицкий С. П., Авдеева Л. И. Линейное и выпуклое программирование. М., Наука, 1966.
12. Калиткин Н. Н. Численные методы. М., Наука, 1978.
13. Карни Ш. Теория цепей. Анализ и синтез. М., Связь, 1973.
14. Корн Г., Корн Т. Справочник по математике. М., Наука, 1968.
15. Мак-Крокен Д., Дори У. Численные методы и программирование на ФОРТРАНе. М., Мир, 1977.
16. Матханов П. П. Основы синтеза линейных электронных цепей. М., Высшая школа, 1976.
17. Нагорный Л. Я. Решение систем линейных уравнений с разреженной матрицей на ЭВМ.—Изв. вузов СССР.—Радиоэлектроника, 1979, № 6.
18. Программное обеспечение ЭВМ МИР-1 и МИР-2. Т1. Численные методы/ В. М. Глушков, И. Н. Молчанов, Б. Н. Брускин и др. Киев, Наукова думка, 1976.
19. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. М., Мир, 1978.
20. Радиоприемные устройства/ Н. В. Бобров, Г. В. Максимов, В. И. Мищурин и др. М., Сов. радио, 1971.
21. Ружинский Л. З. Математическая обработка результатов эксперимента. М., Наука, 1971.
22. Сигорский В. П. Анализ электронных схем. К., Гостехиздат УССР, 1960.

23. Трохименко Я. К. Метод обобщенных чисел и анализ линейных цепей. М., Сов. радио, 1972.
24. Трохименко Я. К. Радиоприемные устройства на транзисторах. К., Техніка, 1972.
25. Трохименко Я. К., Каширский И. С., Ловкий В. К. Проектирование радиотехнических цепей на инженерных ЭЦВМ. К., Техніка, 1976.
26. Трохименко Я. К., Любич Ф. Д. Цифровое моделирование на микрокалькуляторах.—Изв. вузов СССР.—Радиоэлектроника, 1979, № 6.
27. Хеммиг Р. В. Численные методы. М., Наука, 1968.
28. Химмельбау Д. Прикладное нелинейное программирование. М., Мир, 1976.
29. Численные методы/ Н. И. Данилина, Н. С. Дубровская, О. П. Кваша и др. М., Высшая школа, 1976.

СОДЕРЖАНИЕ

Стр.

Предисловие	5
-----------------------	---

Глава 1

Элементы инженерных вычислений

1. Математическое моделирование в инженерных задачах	7
2. Точность и погрешности вычислений	14
3. Моделирование результатов эксперимента	23
4. Преобразования буквенных моделей	37
5. Методы решения уравнений	51
6. Методы решения систем уравнений	62
7. Математическое программирование	73

Глава 2

Вычисления на непрограммируемых микрокалькуляторах

1. Входные языки непрограммируемых микрокалькуляторов	82
2. Особенности вычислений на микрокалькуляторах	90
3. Обработка экспериментальных данных	100
4. Вычисление функций по буквенным моделям	114
5. Решение нелинейных и дифференциальных уравнений	122
6. Решение систем уравнений	134
7. Гармонический и спектральный анализ	147
8. Решение инженерно-экономических задач	159

Глава 3

Программирование микрокалькуляторов

1. Начала программирования	171
2. Особенности программируемых микрокалькуляторов	182
3. Составление программ	196
4. Оптимизация программы	211
5. Типовые арифметические выражения	223
6. Численное интегрирование	236
7. Вычисление специальных функций	242
8. Цифровое моделирование	257

Глава 4

Автоматизация решения типовых задач

1. Решение задач линейной алгебры	270
2. Решение нелинейных уравнений	287
3. Решение алгебраических уравнений	297
4. Решение дифференциальных уравнений	305
5. Автоматизация задач интерполяции и аппроксимации	316
6. Автоматизация спектрального анализа	326
7. Автоматизация статистических расчетов	337
8. Автоматизация задач линейного моделирования	344
9. Особенности автоматизации нелинейного моделирования	360
Приложение	376
Список литературы	380

БИБЛИОТЕКА
ИНЖЕНЕРА
ИНЖЕНЕРНЫЕ
РАСЧЕТЫ
НА МИКРО-
КАЛЬКУЛЯТОРАХ
БИБЛИОТЕКА

Ярослав Карпович Трохименко,
д-р техн. наук

Феликс Дмитриевич Любич,
канд. техн. наук

Редактор
Л. О. Полянская

Оформление художника
Е. В. Попова

Художественные редакторы
Л. А. Дикарев, В. С. Шапошников

Технический редактор
Н. А. Бондарчук

Корректоры Т. Г. Герасименко, Л. В. Ляшенко

Информ. бланк № 1899

Сдано в набор 15.08.79. Подписано в печать 28.03.80. БФ 04193.

Формат 60×90^{1/16}. Бумага типогр. № 1. Гарн. лит.
Печ. выс. Усл. печ. л. 24. Уч.-изд. л. 26. Тираж 20 000
экз. Зак. 9-328. Цена 1 р. 70 к.

Издательство «Техника», 252601, Киев, 1, ГСП, Кре-
щатик, 5.

Книжная фабрика им. М. В. Фрунзе
Республиканского производственного объединения
«Полиграфкнига» Госкомиздата УССР,
Харьков, Донец-Захаржевская, 6/8.