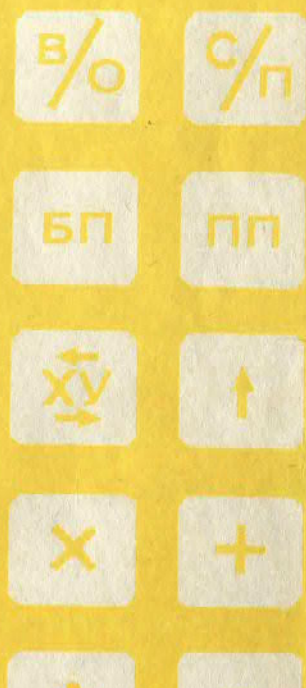
Ш. Е. Цимринг

СПЕЦИАЛЬНЫЕ ФУНКЦИИ
Программы для микрокалькулятора „ЭЛЕКТРОНИКА Б3-21"



ББК 32.937
Ц56
УДК 681.321.0

Цимринг Ш. Е.

Ц56 Специальные функции: Программы для микрокалькулятора «Электроника Б3-21». — М.: Радио и связь, 1983. — 120 с., ил.

35 к.

Содержит программы с подробными комментариями для вычисления значений ряда специальных функций вещественного переменного, а также ряд программ общего назначения (численное решение дифференциальных уравнений первого и второго порядка, вычисление определенных интегралов, в том числе и несобственных, действия над комплексными числами и др.). Каждая программа оформлена в виде удобной таблицы. Приведены контрольные примеры.

Для инженерно-технических работников. Полезна студентам вузов.

ББК 32.937
6Ф7.3

Ц $\dfrac{\mathbf{2405000000\text{-}100}}{\mathbf{046(01)\text{-}83}}$ **104-83**

Рецензенты: д-р техн. наук проф. В. Т. Овчаров; А. В. Ильченко

Редакция литературы по кибернетике и вычислительной технике

ПРЕДИСЛОВИЕ

Зачем нужно вычислять с предварительным программированием функции, если для этой цели служат соответствующие таблицы? Можно указать на два аргумента.

1. Трудно иметь всегда под руками достаточное число подробных таблиц. Время, которое требуется на «добывание» таблиц, подчас перекрывает время вычислений на калькуляторе, даже если библиотека находится в том же здании.

2. Интервал значений аргумента табличных функций, как правило, составляет 1—0,1% от полной величины. Во многих случаях интервалы значительно больше (например, когда аргументом является порядок специальных функций). Время интерполяции по табличным данным при одинаковой точности (5—6 десятичных знаков) существенно превосходит время вычислений на микрокалькуляторе, не говоря о надежности результата.

«Электроника БЗ-21» является программируемым микрокалькулятором. Несмотря на небольшую память (семь регистров, не считая двух операционных регистров и стека) и возможность использования при программировании всего 60 программных шагов, наличие команд условных и безусловных переходов, обращения к подпрограммам, а также очень большая разрядная сетка (10^{-99} ... 10^{99}) создают сравнительно широкие возможности для использования калькулятора в научных и инженерных расчетах.

В предлагаемом сборнике помимо программ специальных функций, число которых можно умножить, приведены некоторые программы общего назначения. К ним относятся интегрирование по Симпсону и по формулам трапеций, нахождение несобственных интегралов (фактически они также являются специальными функциями), интегрирование обыкновенных дифференциальных уравнений и т. д.

Литература, посвященная автоматизации вычислений на микрокалькуляторах, пока немногочисленна. Из имеющихся источников выделяется книга Я. К. Трохименко и Ф. Д. Любича [1], которая содержит более 200 программ различного назначения, главным образом для микрокалькулятора «Электроника БЗ-21».

Задачи настоящей книги уже. Это позволило, однако, расширить по сравнению с [1] раздел «Специальные функции» как за счет числа рассматриваемых функций, так в особенности путем возможно более полного охвата области параметров, определяющих значения функций. Как нам кажется, при этом условии микрокалькулятор сможет принести реальную пользу для лиц, имеющих дело со специальными функциями. Дан также дополнительный материал, относящийся к вычислению определенных интегралов. В особенности это касается несобственных интегралов.

Выбор алгоритмов при составлении программ определяется компромиссом между точностью результата и простотой обращения с программой (минимальный набор исходных данных, отсутствие дополнительных вычислений после окончания счета по программе). Замечательные качества микрокалькулятора «Электроника Б3-21» позволили в большинстве случаев достигнуть компромисса при относительной погрешности менее $1 \cdot 10^{-5}$, т. е. при гарантированных пяти верных цифрах результата.

Поскольку максимальная длина программы на «Электронике Б3-21» составляет 60 шагов, время ввода «самой сложной» программы при внимательном исполнении не превышает 2 мин. Время вычисления одного значения функции обычно лежит в пределах 2 мин. Введенная программа может использоваться для многократных расчетов. В этом случае время ввода программы не имеет значения.

Микрокалькулятор, конечно, нецелесообразно использовать, если под руками имеются приемлемые таблицы. В предлагаемой книге нет программ для вычисления нулей функций, значений специальных функций комплексного аргумента и отсутствует многое другое, что обычно включается в таблицы специальных функций. Поэтому она вместе с микрокалькулятором может рассматриваться и как дополнение к существующим таблицам и справочникам.

В гл. 1 дана информация, необходимая для работы с предлагаемыми программами. Инструкция по эксплуатации прилагается заводом-изготовителем к каждому экземпляру микрокалькулятора. Тем не менее для удобства читателя в гл. 1 даны также основные сведения по выполнению вычислений и программированию на «Электронике Б3-21».

Глава 1

Введение

В § 1.1 даны основные сведения по выполнению вычислений и программированию. В § 1.2 описана организация работы с программами. Материал этого параграфа составлен так, что читатель, имеющий навыки работы на «Электронике Б3-21», может без ущерба для понимания опустить § 1.1.

1.1. ВЫЧИСЛЕНИЕ И ПРОГРАММИРОВАНИЕ НА МИКРОКАЛЬКУЛЯТОРЕ «ЭЛЕКТРОНИКА Б3-21»

Микрокалькулятор производит вычисления непосредственно при нажатии клавиш или автоматически по программе. Вычисления выполняются с целыми или смешанными десятичными числами с плавающей запятой в диапазоне от $\pm 1{,}000000 \cdot 10^{-99}$ до $\pm 9{,}999999 \cdot 10^{99}$.

Ввод чисел или команд, а также действия над числами осуществляются нажатием соответствующих клавиш. Расположение последних на клавиатуре калькулятора показано на рис. 1. Некоторые клавиши имеют двойную или даже тройную символику: один символ нанесен на клавишу, второй находится под клавишей и третий — над ней. Операции, обозначенные символом *над клавишей*, выполняются, когда нажатию на клавишу предшествует нажатие на префиксную клавишу Р (рис. 1). Операции, указанные *под клавишей*, требуют предварительно нажать на префиксную клавишу *F*. Далее *при обозначении операций (команд) используются только символы, нанесенные непосредственно на соответствующие одну или две клавиши.* Например, число π вводится нажатием префиксной клавиши Р, а затем клавиши, на которую нанесен символ $\times$. Поэтому соответствующая команда обозначается как Р $\times$.

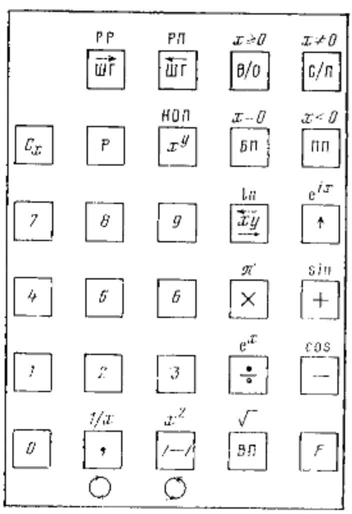


Рис. 1. Клавиатура микрокалькулятора «Электроника Б3-21»

Вводимые числа, результаты вычислений, а также состояние рабочих регистров памяти и программной памяти индицируются на 12-разрядном индикаторе. Характер информации, отражаемой индикатором, существенно разный для двух режимов, в которые может быть введен калькулятор: для рабочего режима и для режима программирования.

При включении калькулятор оказывается всегда в *рабочем режиме.* В этом режиме производятся вычисления или пооперационно нажатием соответствующих клавиш, или автоматически по программе. В *рабочем режиме на индикаторе высвечиваются числа*, которые являются или исходными данными, или результатами вычисления, или содержимыми соответствующих регистров памяти.

Режим программирования устанавливается после нажатия клавиш Р $\overleftarrow{\text{ШГ}}$. В этом режиме программа вводится в память калькулятора, а на индикаторе высвечиваются коды трех следующих одна за другой команд программы и номер последней из них по порядку следования (состояние счетчика адресов команд).

Возвращение из режима программирования в рабочий режим осуществляется нажатием клавиш Р $\overrightarrow{\text{ШГ}}$.

Рабочий режим

В рабочем режиме производятся следующие действия:

—набор чисел на клавиатуре;

—арифметические операции «+», «−», «×», «:»;

—изменение знака числа;

—вычисление встроенных функций $1/x$, x^2, $\sqrt{x}$, $\ln x$, e^x, $\cos x$, $\sin x$, e^{ix}, x^y;

—ввод числа π;

—ввод чисел в регистры памяти и вывод содержимого регистров памяти в операционные регистры и на индикатор, ввод и вывод чисел из стека;

—вычисления по предварительно составленной программе с автоматическим следованием команд или *потактовые*, когда после каждого нажатия клавиши ПП выполняется очередная команда программы и на индикатор выдается соответствующий результат;

—обращение к любой ячейке программной памяти с помощью команды безусловного перехода.

1. *Арифметические операции* и вычисления приведенных встроенных функций выполняются с помощью обоих *операционных регистров* $<X>$ и $<Y>$ или одного из них. *Содержимое регистра* $<X>$ *всегда совпадает с числом на индикаторе в рабочем режиме.* Поэтому в данном режиме набор числа на клавиатуре является одновременно вводом этого числа в регистр $<X>$.

Ввод числа в регистр $<Y>$ может быть осуществлен его перемещением из регистра $<X>$ следующими способами:

нажатием клавиши $\uparrow$. При этом вводимое число сохраняется в регистре $<X>$ и одновременно заполняет регистр $<Y>$;

нажатием клавиши $\overleftarrow{\underrightarrow{xy}}$. Происходит обмен содержимого регистров $<X>$ и $<Y>$. На индикаторе после операции $\overleftarrow{\underrightarrow{xy}}$ высветится предшествующее этой операции состояние регистра $< Y >$. Вторичное нажатие клавиши $\overleftarrow{\underrightarrow{xy}}$ возвращает регистры $< X >$ и $< Y >$ в исходное состояние;

набором нового числа на клавиатуре. «Старое» число перемещается из регистра $< X >$ в регистр $< Y >$. Этот способ можно использовать только в тех случаях, когда после набора «старого» числа произведена какая-либо другая операция. Попытка набора двух чисел подряд воспринимается как набор одного числа.

2. *Набор чисел на клавиатуре* осуществляется нажатием цифровых клавиш 0 . . . 9 в порядке чтения числа. Если число смешанное, то

Разряды индикатора	1	2	3	4	5	6	7	8	9	10	11	12
	–	1	,	7	8	8	6	1	5	–	2	4

Рис. 2. Индикатор микрокалькулятора «Электроника Б3-21»

сначала вводится целая часть, затем запятая (нажатием клавиши «,»), дробная часть и, наконец, знак «—», если число отрицательное (клавиша /—/). Максимальное число разрядов при индикации целых чисел равно 8. При *индикации* смешанных чисел максимальное число разрядов равно 7, так как запятая на индикаторе занимает место одного разряда. Отметим, однако, что память калькулятора всегда содержит 8 разрядов независимо от типа числа. Поэтому для повышения точности *можно набирать и восьмую цифру смешанного числа, хотя эта цифра не высвечивается на индикаторе.*

Чтобы определить восьмую цифру числа a, которое высвечено на индикаторе, можно из него вычесть число b, которое имеет порядок и знак числа a и единственную значащую цифру, равную первой цифре a. Последняя цифра разности $a - b$ и является искомой.

Если необходимо ввести *число в форме с плавающей запятой*, т. е. число с порядком, то сначала следует набрать мантиссу в форме целого или смешанного числа, а после этого ввести порядок. Порядок числа вводится нажатием клавиши ВП, набором однозначного или двузначного числа, характеризующего порядок, и, наконец, если порядок отрицательный, — знака порядка (клавиша /—/). После нажатия клавиши ВП в последних разрядах (11 и 12) индикатора высвечиваются нули. Затем набирается число, совпадающее с порядком, которое сразу индицируется в разрядах 11 и 12. При нажатии на клавишу /—/ в 10-м разряде высвечивается знак порядка «—». На рис. 2 показано расположение числа $-1{,}788615 \cdot 10^{-24}$ на индикаторе.

3. *При выполнении арифметических операций* одно число помещается в регистр $< X >$, а второе — в регистр $< Y >$. Для операций «—» и «:» в регистр $< Y >$ помещается соответственно уменьшаемое и делимое. После введения чисел осуществляется нажатие на клавишу с нужной операцией. На индикаторе и, следовательно, в регистре $< X >$ появляется результат. Число в регистре $< Y >$ при этом сохраняется. Если с ним надо произвести новую арифметическую

операцию, то перед вводом нового числа с клавиатуры надо нажат клавишу C_x для сброса предыдущего числа из регистра $< X >$ Если этого не сделать, то при наборе нового числа в регистре $< Y >$ окажется число, бывшее до этого в регистре $< X >$, т. е. результа предыдущей операции.

Пример. Рассмотрим вычисление a/b. Последовательность операций имее вид: $a \uparrow b \div$. Первая операция — набор числа a на клавиатуре. Это числ оказывается в регистре $<X>$. Нажатие на клавишу $\uparrow$ (вторая операция) вво дит a в регистр $<Y>$, сохраняя a в $<X>$. Затем набираем число b. Это числ замещает a в регистре $<X>$. Итак, в регистр $<X>$ введено число b, а в регист $< Y >$ — число a. Последняя операция $\div$ дает отношение a/b, которое появля ется в регистре $<X>$ и на индикаторе.

4. *Вычисление встроенных функций* $1/x$, x^2, $\sqrt{x}$, $\sin x$, $\cos x$, e^{ix} $\ln x$, e^x требует только ввести x в регистр $< X >$, затем нажать н префиксную клавишу Р или F, а после этого — на клавишу с соответ ствующей операцией. Существенно, что операции $1/x$, x^2, $\sqrt{x}$, $\sin x$ $\cos x$, $\ln x$, e^x не затрагивают содержимого регистра $< Y >$: произ водится только преобразование содержимого регистра $< X >$. Пр операции e^{ix} в регистре $< X >$ оказывается действительная част e^{ix}, т. е. $\cos x$, а в регистре $< Y >$ — мнимая часть, т. е. $\sin x$. *Вво числа* π осуществляется нажатием клавиш Р и $\times$. На индикаторе появ ляется число 3,141592. Отметим, что в память калькулятора засылает ся более точное приближение 3,1415927, с которым и производятс дальнейшие операции. *Для вычисления* x^y число y вводится в регист $< Y >$ указанным способом, а затем число x в регистр $< X >$ После этого осуществляется нажатие клавиши x^y. В регистре $< X >$ появляется результат. В регистре $< Y >$ остается число y. Не до пускаются $x < 0$.

5. *Переполнение* возникает при попытке выполнить некорректну операцию $\ln x$, $\sqrt{x}$, x^y; при $x < 0$, при делении на ноль, а также есл результаты вычислений или вводимые числа превышают $9{,}999999 \times$ $\times 10^{99}$. При переполнении в знаковом (первом) разряде появляется 0 Для восстановления содержимого регистра $< X >$, которое был перед некорректной операцией, достаточно два раза нажать клавиш $\overleftarrow{\underrightarrow{xy}}$.

6. *Работа с регистрами памяти.* Для запоминания исходных ве личин и результатов вычислений могут быть использованы 7 основны регистров памяти, а также в некоторых случаях (см. далее) 6 регист ров памяти стека. Далее основные регистры будут обозначатьс символами $< 2 >$, $< 3 >$, $< 4 >$, $< 5 >$, $<6 >$, $< 7>$, $< 8 >$ Число, находящееся в регистре $< X >$, можно запомнить, введ его в любой из указанных регистров памяти. Для этого надо нажат префиксную клавишу Р, а затем цифровую клавишу, номер которо совпадает с номером регистра.

Число выводится из регистров памяти в $< X >$ нажатием префикс ной клавиши F, а затем цифровой клавиши с номером регистра памят В регистре $< X >$, а следовательно, на индикаторе появится содержи мое соответствующего регистра памяти. Число в регистре памяти сох раняется, т. е. может использоваться многократно, пока не будет заменено новым числом или не будет выключен калькулятор.

Если после вызова числа b из некоторого регистра памяти на клавиатуре набирается другое число b, то a перемещается из $<X>$ в $<Y>$. Это правило упоминалось в п. 1.

Пример. Содержимое a регистра $<2>$ сложить с содержимым b регистра $<3>$, результат разделить на 25 и получившуюся величину запомнить в регистре $<4>$. Выполняем следующие операции:

$$F2 \quad \uparrow \quad F3 \quad + \quad 2 \quad 5 \quad \div \quad \text{P}4.$$

Первые две операции $F2 \uparrow$ вводят число a из $<2>$ в регистры $<X>$ и $<Y>$. Операция $F3$ замещает a в регистре $<X>$ числом b из регистра $<3>$. Операция $+$ дает в регистре $<X>$ сумму $(a+b)$. Далее набираем на клавиатуре 25. Это число оказывается в $<X>$ и на индикаторе, а сумма $(a+b)$ перемещается в $<Y>$. Операция $\div$ дает $(a+b)/25$. Заключительная операция Р4 вводит результат в регистр $<4>$.

7. *Работа со стеком.* Стек — разновидность памяти калькулятора, при использовании которой происходит циклический обмен содержимого регистра с регистрами стека. На рис. 3 регистры стека, обозначаемые цифрами в круглых скобках, для наглядности расположены по окружности. Если нажать клавиши Р /—/ , то произойдет обмен содержимого $<X>$ и двух соседних регистров, например (1) и (7). При этом в регистр (1) будет занесено число из регистра $<X>$, а в $<X>$ вводится число из регистра (7). Если снова нажать клавиши Р /—/, то число из $<X>$ попадет в (7), а в регистре $<X>$ — число из регистра (6). Когда две команды (Р /—/ и Р /—/) следуют одна за другой, то в регистре $<X>$ оказывается число из регистра (6), а начальное содержимое регистра $<X>$ в регистре (1) стека. Содержимое регистра (7) не изменяется. При большем числе повторений команд Р /—/ ситуация аналогична. При нажатии клавиш Р, обмен чисел между регистрами происходит в обратном направлении, т. е. по часовой стрелке (см. рис. 3). Если произвести одну за другой операции Р, и Р /—/, то содержимое всех регистров не изменится.

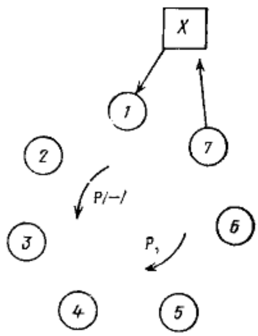


Рис. 3. Обмен числовой информацией между регистром $<X>$ и регистрами стека при команде Р /—/

Отметим, что в отличие от основных регистров памяти нумерация регистров стека на рис. 3 условна: номер 1 присвоен тому регистру стека, в который попадает число из регистра $<X>$ после первой команды Р /—/ или Р,.

Использование стека связано со следующим *ограничением*. Операции $\sin x$, $\cos x$, e^{ix}, $\ln x$, e^x, x^y, как правило, обнуляют числа в одном или нескольких стековых регистрах. *Поэтому между вводом числа в стек и его выводом из стека указанные операции не должны производиться.*

Режим программирования

1. Ввод программы, т. е. последовательности операций (команд), производится в *режиме программирования*. Команды, кодированные двузначными числами (кодами операций), поступают с клавиатуры в программную память калькулятора и там хранятся до выключения калькулятора или записи другой программы. Максимальная емкость программ 60 команд, или «шагов». Каждый шаг снабжен своим *номером*, который называется *адресом* команды. При программировании можно вводить непосредственно (т.е. нажатием соответствующих клавиш) команды, которые калькулятор выполняет без программ (см. предыдущий раздел). Кроме того, имеются команды условных и безусловного переходов, а также команда обращения к подпрограммам, что позволяет организовывать разветвляющиеся программы и многократное прохождение отдельных частей программы (подпрограммы), т. е. циклы. Поэтому, несмотря на сравнительное небольшое число шагов в программе, общее число операций, которое калькулятор может произвести по введенной в него программе, не ограничено.

Адреса команд изображаются двузначными числами, которые высвечиваются в 11-, 12-м разрядах индикатора при записи программы (см. далее). Цифра в младшем разряде пробегает значения от 0 до 5, в старшем — от 0 до 9. Таким образом, первому адресу соответствует число 01, третьему 03, шестому 10, двадцать третьему 35, Адрес 59 изображается числом 95. Последний адрес (60) высвечивается как —0.

2. При вводе программы с 1-го адреса программной памяти переход от рабочего режима к режиму программирования осуществляется нажатием клавиш В/О Р $\overleftarrow{\text{ШГ}}$. На индикаторе в последних двух разрядах (11 и 12) высветится число 00 — исходное состояние счетчика адреса команд.

Пример. Запишем следующую простую программу. Нужно число, введенное в регистр <2> умножить на число из регистра <3> и результат заслать в регистр <6>. Если производить указанные действия в рабочем режиме, то согласно предыдущему разделу потребуется выполнить операции $F2$ ↑ $F3$ × Р6. Рассматривая эту последовательность команд как программу, введем ее в память калькулятора.

Ввод первой команды. Нажимаем клавишу $F2$. В разрядах счетчика адреса число 00 замещается числом 01 — порядковый номер команды 1. В разрядах 1, 2 появляется число 22, которое является кодом операции $F2$.

Вводим следующую команду ↑. В счетчике адреса появляется число 02. В разрядах 1, 2 число 22 замещается числом 06 — кодом операции ↑. Код предыдущей операции перемещается в разряды 4, 5.

После *третьей команды* ($F3$) в адресных разрядах оказывается число 03, в разрядах 1, 2 — число 32, являющееся кодом операции $F3$. Код второй операции перемещается в разряды 4, 5, а код первой операции — в разряды 7, 8.

Вводим четвертую команду. В счетчике адреса появляется число 04, в разрядах 1, 2 — код 26 команды ×. Коды предыдущих двух операций переместятся в разряды 4, 5 и 7, 8, а код операции $F2$ стирается. При записи каждой следующей команды будет высвечиваться в разрядах 1, 2 ее код, а в разрядах 11, 12 — адрес. Коды двух предыдущих команд в порядке убывания адресов высвечиваются в регистрах 4, 5 и 7, 8 соответственно.

Что касается рассматриваемого нами примера, то остается ввести *последнюю операцию* (Р6), а после нее команду автоматического останова (клавиша С/П). Код этой операции 78. На рис. 4 показано состояние индикатора после заключительной команды С/П. В счетчике адреса фигурирует число 10, так как наибольшей цифрой младшего разряда является 5 и за адресом 05 следует 10.

3. Вычисления по введенной программе выполняются следующим образом:

—нажатием клавиш Р $\overrightarrow{\text{ШГ}}$ осуществляется возврат из режима программирования в рабочий режим;

— в основные регистры памяти или в стековые регистры заносятся исходные данные. Для рассматриваемого примера числа, подлежащие перемножению, заносятся в регистры $< 2 >$ и $< 3 >$;

— производится *пуск* программы. Если первая команда программы имеет адрес 01, то пуск осуществляется нажатием клавиш В/О и С/П. По окончании счета (признаком счета является мелькание цифр на

Разряды индикатора	1	2	3	4	5	6	7	8	9	10	11	12
	7	8		6	1		2	6			1	0

Рис. 4. Состояние индикатора после окончания ввода программы

индикаторе) на индикаторе или в заданных программой регистрах памяти окажутся искомые числа. Для нашего примера результат будет на индикаторе и в регистре $< 6 >$.

4. Ввод программы с произвольного адреса (а не с адреса 01), как правило, встречается при вводе подпрограммы или при внесении корректив в набранную программу.

Переход из рабочего режима в режим программирования осуществляется нажатием клавиш БП А Р $\overleftarrow{\text{ШГ}}$. Здесь А — адресная команда. *Адресной командой* будем называть ту команду, код которой совпадает с требуемым адресом перехода. Команда выполняется нажатием одной или двух клавиш. Наименование этих клавиш и является записью адресной команды. Например, адресной команде Р3 соответствует код 31. Следовательно, для перехода в рабочий режим и именно в ячейку 31 должна быть использована система команд БП Р3 Р$\overleftarrow{\text{ШГ}}$. Отметим, что комбинация БП Р3 образует команду безусловного перехода к ячейке 31 (см. далее).

После выполнения указанных команд калькулятор включается в режим программирования и на индикаторе в счетчике адреса высвечивается адрес, который на единицу меньше требуемого. Если теперь приступить к набору программы, то она будет вводиться, начиная с требуемого адреса, который и высветится в счетчике адреса после набора 1-й команды программы.

Как уже упоминалось, в калькуляторе принята нумерация адресов команд, согласно которой в младшем разряде двузначного адреса фигурируют цифры от 0 до 5, а в старшем от 0 до 9. Поэтому при составлении и записи программы удобно команды располагать в ячейках таблицы, содержащей 6 столбцов и 10 строк (рис. 5). Число ячеек в таблице равно 60, т.е. максимальному числу команд. В ячейке, находящейся на пересечении p-й строки и q-го столбца, должна размещаться команда с адресом pq. В ячейках таблицы помещены адресные команды, т. е. наименования клавиш, нажатие которых позволяет «выйти» на нужное место программной памяти при переходе к режиму программирования. Код операций, записанных в каждой ячейке, совпадает с адресом ячейки (и команды). Например, в рабочем режиме нажмем клавиши БП 5 Р $\overleftarrow{\text{ШГ}}$. На индикаторе в счетчике адреса

	0	1	2	3	4	5
0	$\boxed{+}$	P0	F0	P↑	0	F↑
1	↑	P1	F1	$P\overleftrightarrow{xy}$	1	$F\overleftrightarrow{xy}$
2	$\overleftrightarrow{xy}$	P2	F2	P×	2	F×
3	×	P3	F3	P÷	3	F÷
4	÷	P4	F4	P,	4	F,
5	,	P5	F5	P/−/	5	F/−/
6	/−/	P6	F6	PВП	6	FВП
7	ВП	P7	F7	PC_x	7	FC_x
8	C_x	P8	F8	P−	8	F−
9	−	P9	F9	P+	9	F+

Рис. 5. Таблица для записи программы. В ее ячейках помещены адресные команды, необходимые для обращения к данной ячейке при командах безусловных и условных переходов

будет число 53. Таким образом, первая набранная после этого команда окажется в ячейке 54. Здесь 5 — адресная команда для ячейки 54.

Обращаем внимание читателя на то, что последняя (60) команда, адрес которой — 0, помещена в ячейку таблицы 00.

Пуск программы, начинающейся не с ячейки 01, осуществляется нажатием клавиши БП А С/П , где А — адресная команда первой ячейки программы.

Пример. В таблицу на рис. 6 вписана программа вычисления выражения $a + \sqrt{2b^2 + 3c^2}$. Предполагается, что константы a, b, c помещены в регистры <2>, <3> и <4> соответственно. Результат должен быть введен в регистр памяти <8>. Допустим, что программа должна начаться с ячейки 23, которой соответствует адресная команда Р× (см. табл. на рис. 5). Для перехода к режиму программирования нажимаем клавиши БП Р × Р $\overleftarrow{\text{ШГ}}$. После того как программа набрана (в ячейках 23—52 табл. рис. 6 вписаны соответствующие команды*), осуществляется возврат к рабочему режиму, засылка конкрет-

* Команды, записанные в ячейках с 70 по 75, имеют отношение к другому примеру (п. 8).

ных значений констант a, b, c в регистры памяти $<2>$, $<3>$, $<4>$ и пуск программы. Поскольку программа начинается не с ячейки 01, *пуск осуществляется с использованием команды безусловного перехода на адрес первой команды программы.* В рассматриваемом примере мы должны нажать на клавиши БП Р × С/П . Команда БП Р × обеспечивает выход на адрес 23.

5. Ввод корректив в набранную программу. Правильная команда заносится в программную память на место ошибочной. Допустим, что калькулятор был ранее введен в режим программирования, но текущий

	0	1	2	3	4	5
0						
1						
2				F4	F/−/	3
3	×	P5	F3	F/−/	2	×
4	↑	F5	+	FВП	↑	F2
5	+	P8	С/П			
6						
7	−	PВ/0	FC_x	P−		P+
8						
9						

Рис. 6. Таблица программы вычисления значений выражения $a + \sqrt{2b^2+3c^2}$ (ячейки 23—52; величины a, b, c помещаются при вводе исходных данных соответственно в регистры $<2>$, $<3>$, $<4>$; результат засылается в регистр $<8>$). Фрагмент программы, содержащей команду условного перехода, записан в ячейках 70—75

адрес отличается от адреса заменяемой команды. Корректива может вводиться одним из двух способов:

а) адрес заменяемой команды находится недалеко от текущего. Нажимая нужное число раз на клавишу $\overleftarrow{\text{ШГ}}$ или $\overrightarrow{\text{ШГ}}$, производим сдвиг адреса на соответствующее число шагов к началу или к концу программы. После этого вводим правильную команду;

б) адреса текущей и заменяемой команд существенно разнесены. Переходим к рабочему режиму (клавиши Р $\overrightarrow{\text{ШГ}}$) и нажимаем клавиши БП А Р $\overleftarrow{\text{ШГ}}$. Здесь А — адресная команда заменяемой операции. Для отыскания А следует воспользоваться табл. на рис. 5. Далее вводим правильную команду.

6. Лишняя команда исключается вводом в соответствующее место программы команды Р x^y. Ее код равен 39. Надпись НОП над кла-

вишей x^y означает «нет операции». Ввод осуществляется теми же способами, что и внесение корректив — (см. п. 6.).

7. Команда безусловного перехода БП К. Если программа предусматривает после выполнения некоторой операции А переход к операции Б с адресом «б», следует после команды А включить в программу команду безусловного перехода БП К, где К — адресная команда, задающая адрес «б». Напомним, что адресные команды указаны в таблице на рис. 5.

8. Команды условного перехода Р В/О Q, Р С/П Q, Р БП Q, Р ПП Q . Здесь Q — любая адресная команда, задающая адрес перехода (табл. на рис. 5). При выполнении любой из перечисленных команд *содержимое регистра* $<X>$ *проверяется на выполнение условия, указанного над соответствующей клавишей* (для клавиши В/О — условия $x \geqslant 0$; для клавиши С/П — условия $x \neq 0$. БП — условия $x = 0$ и ПП — условия $x < 0$ — см. рис. 1). Если условие *не выполняется*, то произойдет переход на команду программы, адрес которой задает команда Q. Если условие *выполняется*, то программа будет продолжаться, начиная с команды, которая следует *сразу же за* Q.

В ячейках 70 — 75 таблицы на рис. 6 записан фрагмент программы, содержащей команду условного перехода Р В/О F C_x . Если после выполнения операции «—» (ячейка 70) в регистре $<X>$ окажется отрицательное число, то следующей будет выполнена операция вычисления синуса этого числа Р + (ячейка 75). При положительном или равном нулю результате операции «—» будет вычислен косинус (команда Р — в ячейке 73).

9. Обращение к подпрограмме производится нажатием клавиш ПП и Е, где Е — наименование клавиш, совпадающее с адресной командой начала подпрограммы. *Последней в подпрограмме* должна быть обязательно команда возврата В/О. После окончания вычислений по подпрограмме вызывающая программа (или подпрограмма) продолжается сразу же за командой обращения к подпрограмме. Обращение к подпрограмме может производиться многократно из основной программы или подпрограммы. В последнем случае согласно заводскому описанию наибольшая глубина обращения равна 5.

Отметим, что в данном калькуляторе команда обращения к подпрограммам не выполняется, если обращение осуществляется с заштрихованных ячеек таблицы на рис. 5.

10. Проверка программы может производиться двумя способами:

а) контроль кодов программы. Последовательно нажимая клавишу $\overrightarrow{\text{ШГ}}$, начиная с первой команды программы. При каждом нажатии происходит сдвиг на один шаг в порядке возрастания адресов команд. В разрядах 1,2 появляется двузначный код, который должен совпадать с правильным кодом текущей команды. Коды всех операций и адресные коды приведены в табл. 1;

б) потактовое прохождение программы. Переходим к рабочему режиму и вводим исходные данные, как при автоматическом счете по программе. Нажимаем на клавишу В/О, если программа введена с

Таблица 1. **Коды операций, адресные коды и соответствующие команды**

Код	01	02	03	04	05	06	11	12	13	14	15	16
Команда	Р0	$F0$	Р↑	0	F↑	↑	Р1	$F1$	Р$\overset{\leftarrow}{\underset{\rightarrow}{xy}}$	1	$F\overset{\leftarrow}{\underset{\rightarrow}{xy}}$	$\overset{\leftarrow}{\underset{\rightarrow}{xy}}$
Код	21	22	23	24	25	26	31	32	33	34	35	36
Команда	Р2	$F2$	Р×	2	F×	×	Р3	$F3$	Р÷	3	F÷	÷
Код	41	42	43	44	45	46	51	52	53	54	55	56
Команда	Р4	$F4$	Р,	4	F,	,	Р5	$F5$	Р/−/	5	F/−/	/−/
Код	61	62	63	64	65	66	71	72	73	74	75	76
Команда	Р6	$F6$	РВП	6	FВП	ВП	Р7	$F7$	РC_x	7	FC_x	C_x
Код	81	82	83	84	85	86	91	92	93	94	95	96
Команда	Р8	$F8$	Р−	8	F−	−	Р9	$F9$	Р+	9	F+	+
Код	38	48	58	68	78	39	49	59	69	79		
Команда	x^y	В/О	БП	ПП	С/П	Рx^y	РВ/О	РБП	РПП	РС/П		

ячейки 01, или нажимаем на клавиши БП А, где А — адресная команда начала программы. А находится по табл. на рис. 5.

Далее нажимаем на клавишу ПП. После каждого нажатия должна выполняться очередная команда программы. Сверяем на каждом шаге содержимое индикатора с требуемым промежуточным результатом вычислений.

При *отладке* программы целесообразно использовать параллельно оба варианта. При вводе *отлаженной* программы достаточен вариант а).

11. Оптимизация программ. Малое число команд (60) и регистров памяти ставят возможность программирования многих задач в прямую зависимость от того, насколько экономично в смысле числа используемых шагов и регистров памяти составлена программа. Здесь многое зависит от рационального построения исходной формулы. Рассмотрим в связи с этим два примера.

Пример 1. Вычислить sh x. Аргумент x предполагается введенным в регистр $<2>$. Воспользуемся формулой sh $x = (e^x - e^{-x})/2$. Программа имеет вид $F2$ /—/ Р÷ Р3 $F2$ Р÷ ↑ $F3$ — 2 ÷ С/П .

Здесь операции $F2$ /—/ Р÷ Р3 дают e^{-x} и засылают эту величину в регистр $<3>$. Команды $F2$ Р÷ ↑ дают e^x в регистре $<Y>$ и т. д. Всего программа требует 12 команд и два регистра памяти.

Если представить гиперболический синус в виде sh $x = (e^x - 1/e^x)/2$, то программа примет следующий вид:

$F2$ Р÷ ↑ F, — 2 ÷ С/П .

Длина программы сократилась до 8 команд, и отпала необходимость в регистре памяти для хранения промежуточного результата.

	0	1	2	3	4	5
0	$\boxed{}$	F2 12345	Р4 12345	F3 67890	Р5 67890	F4 12345
1	Р6 12345	F5 67890	Р7 67890	F6 12345	↑ 12345	F7 67890
2	$\overleftarrow{\overrightarrow{xy}}$ 12345	÷ 5,499392	Р8 5,499392	F, $1{,}818382 \cdot 10^{-1}$	↑ $1{,}818382 \cdot 10^{-1}$	F8 5,499392
3	x^y 1,363381	× $2{,}479149 \cdot 10^{1}$	F8 5,499392	+ 5,681230	↑ 5,681230	F6 12345
4	$\overleftarrow{\overrightarrow{xy}}$ 5,681230	— 12339,31	Р$\overleftarrow{\overrightarrow{xy}}$ 9,420545	FВП 3,069290	Р+ 0,072239	Р— 0,997391
5	Р÷ 2,711201	Р× 3,141592	× 38782,96	Р5 38782,96	РПП 38782,96	F6 38782,96
6	1 (1)	С/П (1)	РБП 38782,96	ВП 38782,96	2 (2)	С/П (2)
7	C_x 0	РС/П 0	FC_x 0	3 (3)	С/П (3)	1 1
8	/—/ −1	РВ/О −1	F— −1	4 (4)	С/П (4)	БП −1
9	Р+ −1	0 (0)	С/П (0)	5 5	С/П 5	

Рис. 7. Программа для проверки микрокалькулятора

Пример 2. Вычислить значение дробно-линейной функции $f(x) = (x - 5)/(x + 2)$. Аргумент x помещен в регистр $<7>$.

Первый вариант программы $F7$ 2 + Р2 $F7$ 5 — ↑ $F2$ ÷ С/П .

Представим $f(x)$ в виде $f(x) = (x - 5)/[(x - 5) + 7]$. Соответствующая программа $F7$ 5 — 7 + ÷ С/П короче первой на 4 команды и не нуждается в регистре памяти для хранения промежуточного результата.

12. Проверка микрокалькулятора. Если правильность работы микрокалькулятора вызывает сомнения, то можно использовать следующую тестовую программу, которая позволяет обнаружить дефектную операцию, если она имеется. *Программа не распространяется на команды обращения к подпрограмме*. В табл. на рис. 7 приведена тестовая программа. При ее использовании выполняются следующие действия:

— ввод числа 12345 в регистр $< 2 >$;

— ввод числа 67890 в регистр $< 3 >$;

— пуск В/О С/П .

Если все арифметические операции, операции вычисления встроенных функций, операции условных и безусловных переходов и операции за-

ылки и выборки чисел из регистров памяти правильны, то после оста-
ова на индикаторе должно быть число 5, а в регистре < 5 > —
исло 38782, 96.

Если в регистре < 5 > будет иное число, то это означает, что
акая-то арифметическая операция, операция вычисления встроенной
ункции или засылки-выборки производится неправильно. В каждой
чейке помещены числа, которые должны получиться после команды,
писанной в ячейке. Поэтому при потактовом прохождении програм-
ы легко обнаружить дефектную команду.

Наличие на индикаторе после останова вместо числа 5 одного из
исел 1, 2, 3, 4 и 0 указывает на неисполнение команд безусловного
ерехода или условных. Если на индикаторе число

1 — не проходит команда Р ПП;
2 — не проходит команда Р БП;
3 — не проходит команда Р С/П;
4 — не проходит команда Р В/О;
0 — не проходит команда БП.

Ячейки, в которых числа записаны в скобках, обходятся при нор-
альной работе калькулятора. Поэтому при отсутствии дефектов со-
ответствующие команды не исполняются и указанные в этих ячейках
исла пропускаются.

1.2. ОРГАНИЗАЦИЯ ПРОГРАММ

1. Процесс вычисления по программе состоит из следующих этапов:
а) переход в режим программирования;
б) ввод программы в память калькулятора;
в) переход в рабочий режим;
г) занесение исходных данных;
д) пуск программы.

ри многократных расчетах повторяются только этапы г) и д).

Программы специальных функций даются в отлаженном виде, и
роцесс ввода программы состоит в нажатии клавиш, указанных в со-
ответствующей таблице. Программы вычисления определенных интег-
лов, решения дифференциальных уравнений и некоторые другие (см.
3.3, 3.4) нуждаются в составлении и отладке пользователем подпро-
амм расчета подынтегрального выражения, правой части дифферен-
иального уравнения, разрешенного относительно производной, и т. д.

2. Каждая программа оформлена в виде таблицы, содержащей 6
олбцов и 10 строк. 60 ячеек таблицы соответствуют полной програм-
ой памяти микрокалькулятора. На рис. 8 показана подобная таб-
ица, в которой записана программа вычисления суммы $(a + b)$.
редполагается, что значения констант a и b будут помещены соответ-
венно в регистры памяти < 2 > и < 4 >, а результаты сложения —
регистр памяти < 6 >. В левом верхнем углу ячеек таблицы
исаны команды, а в правом нижнем углу — их числовые коды.

Программа записывается *построчно* в порядке возрастания адре-
в команд, начиная с адреса 01. *Адрес команды* — двузначное число,
младшем разряде которого стоит номер столбца таблицы, а в стар-

нем — номер строки. *Ячейка 00 таблицы используется для записи по-следней команды* (60), если программа имеет максимальную дли (60 шагов).

3. Переход в режим программирования из рабочего режима про-водится нажатием клавиш БП Е Р $\overleftarrow{\text{ШГ}}$. Здесь все обозначен кроме Е, соответствуют надписям на клавишах. Е — адресная кома начала программы. *Адресная команда* — обозначение одной или д клавиш, нажатие на которые вырабатывают числовой код, равн адресу нужной команды. Адресные команды для всех адресов (яч таблицы типа приведенной на рис. 8) даны в табл. на рис. 5.

	0	1	2	3	4	5
0	$\boxed{}$	F2 27	↑ 0b	F4 42	+ 96	F6 61
1	С/П 78					
2						
3						
4						
5						
6						
7						
8						
9						

Рис. 8. Таблица для записи программы

В большинстве случаев программа начинается с первого адре т. е. первая команда записывается в ячейку 01. В этом случае перех к режиму программирования осуществляется нажатием клавиш В Р $\overleftarrow{\text{ШГ}}$.

4. Ввод очередной команды программы осуществляется нажати одной или двух клавиш, обозначенных в левом верхнем углу кажд ячейки таблицы. Знаки, которые там фигурируют, нанесены на со ветствующие клавиши калькулятора (рис. 1). Ввод производится *строчно в* порядке возрастания адресов команд.

После нажатия очередной клавиши в крайних правых (11, 12) р рядах индикатора (см. рис. 2) должен появиться адрес команды (яч ки таблицы). На левом краю индикатора (разряды 1, 2) высвечивает двузначный код команды. Коды команд записаны в правом нижн краю каждой ячейки. Таким образом, *если текущая команда набра правильно* и не было пропущенных или лишних команд, содержим рядов 1, 2 индикатора должно совпадать с числом в правом нижнем лу ячейки, а содержимое разрядов 11, 12 — с адресом ячейки.

Если команда набрана ошибочно, то следует возвратить программы зад на один или несколько шагов (соответствующее число нажатий авиши $\overleftarrow{\text{ШГ}}$) и повторить набор команды.

5. Переход в рабочий режим после окончания ввода программы уществляется нажатием клавиш Р $\overrightarrow{\text{ШГ}}$.

6. Ввод исходных данных производится путем засылки соответст-ющих чисел в регистры памяти и, если это требуется, в регистры ека. Информация о распределении памяти дается таблицами, типа

с. 9. Таблица расположения исход- х данных и результатов вычисле- ний в регистрах памяти

иведенной на рис. 9, где показаны данные для программы вычисле- я ($a+b$) (см. табл. на рис. 8). В описании к каждой программе мещена подобная таблица, на которой показаны те регистры памяти, которые должны быть внесены исходные данные, и регистры, в торые будут помещены результаты счета при автоматическом про- ждении программы. В описаниях к соответствующим программам казаны также свободные регистры памяти, которые могут быть пользованы при программировании правых частей дифференциаль- ых уравнений или подынтегральных функций (они выделены зна- м ∼).

7. Способ пуска программы, т. е. включения микрокалькулятора в жим автоматического счета по программе, зависит от того, с какого реса она начинается. В подавляющем числе случаев адресом первой манды является 01. Тогда для *пуска* нужно нажать клавиши /О С/П . Если адресом первой команды является другая ячейка некоторой *адресной командой* Е, то *пуск* выполняется по команде П Е С/П .

8. В описании к каждой программе приводятся определение функ- ий и информация о достигаемых точности и времени счета. Определе- ия и обозначения специальных функций, как правило, соответствует блицам И. С. Градштейна и И. М. Рыжика. [2].

В описаниях даны также один или несколько численных примеров. *астоятельно рекомендуется* после работы набора нужной программы рорешать» хотя бы один из примеров. Это занимает несколько минут. сли программа и исходные данные введены без ошибок, то *все разряды* ответствующих чисел должны *совпасть*. Тогда можно с должным верием отнестись и к «своим» результатам. Подготовку исходных анных, если она связана с предварительными вычислениями, жела- ельно производить на микрокалькуляторе с последующим перено- м получаемых чисел в регистры памяти командами ввода. В противном учае возможна потеря точности.

Глава 2

Программы вычисления специальных функций

2.1. ОБРАТНЫЕ ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ

ОПРЕДЕЛЕНИЯ

$$\alpha = \arcsin x, \text{ если } \sin\alpha = x \text{ и } -\frac{\pi}{2} \leqslant \alpha \leqslant \frac{\pi}{2} \text{ при } -1 \leqslant x \leqslant 1$$

$$\beta = \arccos x, \text{ если } \cos\beta = x \text{ и } 0 \leqslant \beta \leqslant \pi \text{ при } 1 \geqslant x \geqslant -1,$$

$$\gamma = \operatorname{arctg} x, \text{ если } \operatorname{tg}\alpha = x \text{ и } -\frac{\pi}{2} \leqslant \gamma \leqslant \frac{\pi}{2} \text{ при } -\infty < x < \infty$$

Метод расчета

В основном используется метод итераций. Пусть $\alpha_1 = \alpha_0 +$ где $\Delta\alpha \ll 1$, и т. д. Из приведенных определений для трех указан функций α, β, γ получаем соответственно

$$\Delta\alpha = (x - \sin\alpha_0)/\cos\alpha_0, \tag{2.1}$$

$$\Delta\beta = (\cos\beta_0 - x)/\sin\beta_0, \tag{2.2}$$

$$\Delta\gamma = (x - \operatorname{tg}\gamma_0)/(1 + x\operatorname{tg}\gamma_0). \tag{2.3}$$

Итерационные схемы строим следующим образом. Задаем α_0, β_0, и из (2.1)—(2.3) получаем первую итерацию $\alpha_1 = \alpha_0 + \Delta\alpha$; $\beta_1 = \beta_0 + \Delta\beta$; $\gamma_1 = \gamma_0 + \Delta\gamma$. Заменяя α_0 на α_1 и т. д. в (2.1)—(2.3), нахо новые значения $\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$ и т. д.

Эта процедура приводит, однако, к значительным относительн погрешностям при малых значениях искомых функций за счет погр ности округления. Поэтому указанный алгоритм дополняется пере дом к представлению функций отрезками соответствующих степен рядов. Во всех случаях достаточно использовать первые два чле

$$\alpha = \arcsin x \approx x + x^3/6 \quad \text{при} \quad |x| < 0{,}03;$$

$$\beta = \arccos x \approx \sqrt{(1 - x)(7 - x)/3} \quad \text{при} \quad x > 0{,}99;$$

$$\gamma = \operatorname{arctg} x \approx x - x^3/3 \quad \text{при} \quad |x| < 0{,}01.$$

В качестве начального значения в итерационных процеду для $\arcsin x$ используется $\alpha_0 = x + x^3/6$; для $\arccos x$ $\beta_0 = \sqrt{(1 - x)(7 - x)/3}$ и для $\operatorname{arctg} x$ $\gamma_0 = \pm 1$ (верхний знак при $x >$ и нижний — при $x < 0$).

ПРОГРАММА 1

$\alpha = \arcsin x$

	0	1	2	3	4	5
0		F2 22	F/−/ 55	6 64	÷ 36	1 14
1	+ 96	↑ 06	F2 22	× 26	P3 31	F/−/ 55
2	y 94	P4 41	ВП 66	4 44	/−/ 56	− 86
3	PБ/0 49	РВП 63	F2 22	↑ 06	F3 32	P+ 93
4	− 86	↑ 06	F3 32	P− 83	÷ 36	↑ 06
5	F3 32	+ 96	P3 31	F4 42	1 14	− 86
6	P4 41	РПП 69	F3 32	F3 32	1 14	8 84
7	0 04	× 26	↑ 06	P× 23	÷ 36	P4 41
8	С/П 78					
9						

		$(\alpha)_{\text{рад}}$	$(\alpha)_{\text{град}}$				
гистры мяти	2	3	4	5	6	7	8
	x						

Порядок работы

1. *Ввод* x в <2>.
2. Пуск В/О С/П.
3. *Результаты*: Значения α в радианах содержатся в регистре <3>. Значения α в градусах — в регистре <4> и на индикаторе.

Относительная погрешность меньше $5 \cdot 10^{-5}$ при $|x| \leqslant 0{,}999999$; 10^{-5} при $|x| \leqslant 0{,}9999$ (достоверны 5 значащих цифр).

Время счета примерно 1,5 мин.

Примеры:

arc sin (0,031) = 0,03100496 = 1,776453°. Табличное значение 0,03100496 ...;

arc sin ($\sqrt{3}/2$) = 1,0471975 = 59,999996°;

arc sin (— 0,99999) = — 1,5663295 = — 89,744068°. Табличное значение 89,74376°.

ПРОГРАММА 2

$\beta = \arccos x$

	0	1	2	3	4	5
0	$\boxed{}$	F2 22	1 14	− 86	6 64	− 86
1	× 26	3 34	÷ 36	FВП 65	Р3 31	F2 22
2	3 34	Р− 83	+ 96	РПП 69	F6 62	9 94
3	Р4 41	F3 32	Р− 83	↑ 06	F2 22	− 86
4	↑ 06	F3 32	Р+ 93	÷ 36	↑ 06	F3 32
5	+ 96	Р3 31	F4 42	1 14	− 86	Р4 41
6	РПП 69	Р3 31	F3 32	1 14	8 84	0 04
7	× 26	↑ 06	Р× 23	÷ 36	Р4 41	С/П 78
8						
9						

		(β) рад	(β) град			
Регистры памяти	2	3	4	5	6	7
	x					

Порядок работы

1. *Ввод* x в<2>.
2. Пуск В/О С/П.
3. *Результаты*: Значения β в радианах — в регистре <3>. Значения β в градусах — в регистре <4> и на индикаторе. *Относительная погрешность* меньше $1 \cdot 10^{-6}$.

Примеры:

arc cos (0,5) = 1,047197 = 59,99999°;

arc cos (0,9999) = $1{,}414225 \cdot 10^{-2}$ = 0,8102913°. **Табличное зна**
$1{,}414222 \cdot 10^{-2}$;

arc cos (— 0,999999) = 3,140205 = 179,9205°. **Табличное значение 3,1402**

Время счета 1,5 мин.

ПРОГРАММА 3

$\gamma = \operatorname{arc\,tg} x$

	0	1	2	3	4	5
0	$\boxed{\text{С/П} \; 78}$	F2 22	F/−/ 55	3 34	÷ 36	1 14
1	− 86	/−/ 56	↑ 06	F2 22	× 26	P3 31
2	F/−/ 55	9 94	P4 41	/−/ 56	P+ 33	− 86
3	PВ/0 49	8 84	F2 22	↑ 06	F/−/ 55	FВП 65
4	÷ 36	P3 31	F3 32	P↑ 03	÷ 36	↑ 06
5	F2 22	− 86	/−/ 56	P, 43	F2 22	× 26
6	1 14	+ 96	↑ 06	P/−/ 53	$\overrightarrow{xy}$ 16	÷ 36
7	↑ 06	F3 32	+ 96	P3 31	F4 42	1 14
8	− 86	P4 41	PПП 69	F4 42	F3 32	1 14
9	8 84	0 04	× 26	↑ 06	P× 23	÷ 36

		(γ) рад					
истры яти	2	3	4	5	6	7	8
	x						

Порядок работы

1. *Ввод* x в <2>.
2. *Пуск* В/О С/П.
3. *Результаты*: Значения γ в радианах — в регистре <3>. Значения в градусах — на индикаторе.

Относительная погрешность меньше $5 \cdot 10^{-6}$.

Время счета примерно 1,5 мин.

Примеры:

$\operatorname{arc\,tg}(1 \cdot 10^5) = 1{,}570786 = 89{,}99942°$. Табличное значение 1,570786;
$\operatorname{arc\,tg}(-10) = -1{,}471127 = -84{,}28941°$. Табличное значение — 1,471127;
$\operatorname{arc\,tg}(\sqrt{3}/3) = 0{,}5235988 = 30{,}00000°$;
$\operatorname{arc\,tg}(0{,}012) = 1{,}199947 \cdot 10^{-2} = 0{,}687519°$. Табличное значение $1{,}199942 \times 10^{-2}$.

2.2. ГИПЕРБОЛИЧЕСКИЕ И ОБРАТНЫЕ ГИПЕРБОЛИЧЕСКИЕ ФУНКЦИИ

ОПРЕДЕЛЕНИЯ

$$\operatorname{sh} x = \frac{e^x - e^{-x}}{2}; \quad \operatorname{ch} x = \frac{e^x + e^{-x}}{2}; \quad \operatorname{th} x = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

arsh x, arch x, arth x — обратные функции. Легко показать, ч

$$\operatorname{arsh} x = \ln\left(x + \sqrt{x^2 + 1}\right);$$

$$\operatorname{arch} x = \ln\left(x + \sqrt{x^2 - 1}\right), \quad x \geqslant 1;$$

$$\operatorname{arth} x = \ln\left[(1 + x)/(1 - x)\right]/2, \quad x < 1.$$

ПРОГРАММА 4

ФУНКЦИИ sh x, ch x, th x, arsh x, arch x, arth x

	0	1	2	3	4	5
0	$\boxed{\text{С/П} \quad 78}$	F2 22	P÷ 33	↑ 06	F, 45	P7 71
1	+ 96	2 24	÷ 36	P4 41	↑ 06	F7 72
2	− 86	P3 31	↑ 06	F4 42	÷ 36	P5 51
3	F2 22	F/−/ 55	1 14	+ 96	FВП 65	↑ 06
4	F2 22	+ 96	P$\underset{\rightarrow}{xy}$ 13	P6 61	F2 22	1 14
5	− 86	PПП 69	FC_x 75	0 04	P7 71	F2 22
6	1 14	+ 96	2 24	− 86	/−/ 56	÷ 36
7	P$\underset{\rightarrow}{xy}$ 13	2 24	÷ 36	P8 81	С/П 78	0 04
8	0 04	P8 81	F2 22	F/−/ 55	1 14	− 86
9	FВП 65	↑ 06	F2 22	+ 96	P$\underset{\rightarrow}{xy}$ 13	P7 71

Регистры памяти	2	3	4	5	6	7	8
	x	sh x	ch x	th x	arsh x	arch x	ar

Порядок работы

1 *Ввод* x в <2>.

2 *Пуск* В/О С/П.

3. *Результаты*: sh x в <3>; ch x в <4>; th x в <5>; a
в <6>; arch x в <7>; arth x в <8>. Наличие 0 в <7> озна
что arch x не существует ($x < 1$). Точно также наличие 00 в <8>
зывает на то, что $x \geqslant 1$ и arth x не существует.

Относительная погрешность $< 1 \cdot 10^{-4}$.

Время счета примерно 15 с.

Пример. Вычислить функции при $x = 2{,}995$.

sh x = 9,967659; ch x = 10,01769; th x = 0,9950051;

arsh x = 1,816862; arch x = 1,760974; arth x = 00 (не существует).

2.3. ИНТЕГРАЛЬНАЯ ПОКАЗАТЕЛЬНАЯ ФУНКЦИЯ, ИНТЕГРАЛЬНЫЙ СИНУС, ИНТЕГРАЛЬНЫЙ КОСИНУС

ОПРЕДЕЛЕНИЯ

Интегральная показательная функция

$$\mathrm{Ei}\,(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t}\,dt,\ x < 0; \tag{2.4}$$

$$\mathrm{Ei}\,(x) = -\mathrm{VP}\int_{-x}^{\infty} \frac{e^{-t}}{t}\,dt,\ x > 0. \tag{2.5}$$

Здесь VP — главное значение интеграла по Коши, т. е.

$$\mathrm{Ei}\,(x) = -\lim_{\varepsilon\to 0}\left[\int_{-x}^{-\varepsilon} \frac{e^{-t}}{t}\,dt + \int_{\varepsilon}^{\infty} \frac{e^{-t}}{t}\,dt\right],\ x > 0,\ \varepsilon > 0.$$

Интегральный синус

$$\mathrm{Si}\,x = \int_{0}^{x} \frac{\sin x}{x}\,dx. \tag{2.6}$$

Интегральный косинус

$$\mathrm{Ci}\,(x) = \int_{0}^{x} \frac{\cos t - 1}{t}\,dt + C + \ln x, \tag{2.7}$$

где $C = 0{,}5772157\ldots$ — постоянная Эйлера.

МЕТОД РАСЧЕТА

Ei (x) вычисляется с помощью рядов по формуле [2]

$$\mathrm{Ei}\,(x) = C + \ln|x| + \sum_{n=1}^{\infty} \frac{x^n}{n\cdot n!}. \tag{2.8}$$

Формула (2.8) при $x < 0$ пригодна для относительно малых $|x|$. При больших $|x|$ члены ряда по модулю велики и при их суммировании имеет место потеря точности за счет погрешности округления, так как ряд является знакочередующимся (см., например, [3]). Оценки показывают, что для получения точности не хуже 10^{-8} следует брать $x > -5$. Для $x > 0$ указанное ограничение отпадает и (2.8) можно в принципе

использовать до весьма больших x. При увеличении x растет число у
тываемых членов ряда и время счета (см. описание к программе

Si (x) и Ci (x) вычисляются непосредственно по (2.6) и (2.7) ин
грированием по формуле Симпсона (см. § 3.1). Число шагов вычис
ется в программе из условия $N \sim 8x$.

ПРОГРАММА 5

ИНТЕГРАЛЬНАЯ ПОКАЗАТЕЛЬНАЯ ФУНКЦИЯ Ei (x)

	0	1	2	3	4	5
0	$\boxed{}$	1 14	P4 41	0 04	P3 31	P5 51
1	1 14	ВП 66	8 84	/−/ 56	P7 71	F3 32
2	1 14	+ 96	P3 31	F4 42	↑ 06	F2 22
3	× 26	↑ 06	F3 32	÷ 36	P4 41	↑ 06
4	F3 32	÷ 36	↑ 06	F5 52	+ 96	P5 51
5	÷ 36	F/−/ 55	FВП 65	↑ 06	F7 72	− 86
6	PПП 69	F$\underrightarrow{xy}$ 15	F2 22	F/−/ 55	FВП 65	P$\underrightarrow{xy}$ 13
7	↑ 06	F5 52	+ 96	0 04	, 46	5 54
8	7 74	7 74	2 24	1 14	5 54	7 74
9	+ 96	P5 51	С/П 78			

Регистры памяти	2	3	4	5	6	7	8
	x			Ei (x)			

Порядок работы

1. *Ввод* x в $<2>$.
2. *Пуск* В/О С/П.
3. *Результат*: Ei (x) выдается на индикатор и хранится в <5

Относительная погрешность при $x \geqslant -5$ меньше $1 \cdot 10^{-3}$; п
$x \geqslant -1{,}5$ меньше $1 \cdot 10^{-4}$; при $x \geqslant 1$ меньше $1 \cdot 10^{-5}$ (единицы в
значащей цифре).

Время счета $t \approx (1 + |x|/5)$ мин.

Примеры:

Ei (— 4,5) = — 0,0020738. Табличное значение — 0,002073;

Ei (1) = 1,8955118. Табличное значение 1,8951178;

$[xe^{-x}\mathrm{Ei}(x)]_{x=100}$ = 1,010204. Табличное значение 1,0102062 [4];

Ei (100) = $2{,}715548 \cdot 10^{41}$.

ПРОГРАММА 6

ИНТЕГРАЛЬНЫЙ СИНУС Si (x)

	0	1	2	3	4	5
0	$\boxed{}$	F8 82	6 64	÷ 36	↑ 06	F7 72
1	÷ 36	P3 31	2 24	× 26	P4 41	F8 82
2	+ 96	P2 21	1 14	P5 51	ПП 68	F7 72
3	ПП 68	F7 72	+ 96	+ 96	+ 96	P5 51
4	F2 22	↑ 06	F4 42	− 86	РПП 69	F6 62
5	F5 52	↑ 06	F3 32	× 26	3 34	÷ 36
6	P5 51	С/П 78	ПП 68	F7 72	+ 96	P5 51
7	БП 58	× 26	F2 22	↑ 06	F3 32	− 86
8	P2 21	P+ 93	↑ 06	F2 22	÷ 36	↑ 06
9	F5 52	+ 96	P5 51	В/О 48		

Регистры памяти	2	3	4	5	6	7	8
				Si (x)		A (x)	x

Порядок работы

1. *Ввод аргумента* x в <8>.
2. *Ввод целой части* аргумента A (x) в <7> (округление *обязательно* в большую сторону).
3. *Пуск* В/О С/П.
4. *Результат*: Si (x) выдается на индикатор и содержится в регистре <5>.

Относительная погрешность меньше: $1 \cdot 10^{-5}$.

Время счета примерно (x/2) мин.

Пример.

Si (3,7) = 1,808622. Табличное значение 1,80862168...

ПРОГРАММА 7
ИНТЕГРАЛЬНЫЙ КОСИНУС Ci (x)

	0	1	2	3	4	5
0	$\boxed{\text{В/О}}$ 48	F8 82	8 84	÷ 36	↑ 06	F7 72
1	÷ 36	P3 31	2 24	× 26	P4 41	F8 82
2	+ 96	P2 21	ПП 68	PCx 73	ПП 68	PCx 73
3	+ 96	+ 96	+ 96	P5 51	F2 22	↑ 06
4	F4 42	− 86	РПП 69	РВП 63	F5 52	↑ 06
5	F3 32	× 26	3 34	÷ 36	↑ 06	F8 82
6	P $x \rightleftarrows y$ 13	+ 96	С/П 78	ПП 68	PCx 73	+ 96
7	P5 51	БП 58	2 24	F2 22	↑ 06	F3 32
8	− 86	P2 21	P− 83	1 14	− 86	↑ 06
9	F2 22	÷ 36	↑ 06	F5 52	+ 96	P5 51

Регистры памяти	2	3	4	5	6	7	8
				0		$A(x)$	x

Порядок работы

1. *Ввод* x в <8>.
2. *Ввод* целой части аргумента A (x) в <7> (округление проводить в большую сторону).
3. *Ввод* 0 в <5>.
4. *Пуск* В/О С/П .
5. *Результат*: Для получения Ci (x) к величине, которая выдася в конце счета на индикатор, прибавить постоянную Эйлера C = 0,5772157. В регистре <5> хранится величина $24 \cdot (\mathrm{Ci}(x) - C - \ln x) \cdot A(x)/x$.

Абсолютная погрешность меньше $3 \cdot 10^{-6}$.

Время счета примерно x мин.

Примеры:

Ci (9,9) = — 0,0367656. Табличное значение — 0,03676395...;
Ci (2,3) = 0,3471737. Табличное значение 0,34717561.

2.4. ГАММА-ФУНКЦИЯ, ЛОГАРИФМИЧЕСКАЯ ПРОИЗВОДНАЯ ГАММА-ФУНКЦИИ (ДИГАММА-ФУНКЦИЯ), ПОЛИГАММА-ФУНКЦИ

ОПРЕДЕЛЕНИЯ

Гамма-функция для $x > 0$

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt. \quad (2$$

ри $x < 0$ и в комплексной области гамма-функция определяется как налитическое продолжение (2.9). Для продолженной функции применимо асимптотическое разложение Стирлинга

$$\Gamma(z) = \sqrt{\frac{2\pi}{z}}\, e^{-z} z^z \left(1 + \frac{1}{12z} + \ldots\right). \qquad (2.10)$$

оскольку известное рекуррентное соотношение

$$\Gamma(z+1) = z\Gamma(z) \qquad (2.11)$$

акже справедливо во всей области, то (2.10) и (2.11) можно в совокупости рассматривать как определение $\Gamma(z)$, справедливое, в частности, ри всех вещественных z.

Логарифмическая производная Γ-*функции* (*дигамма-функция*)

$$\psi(x) \equiv \frac{d}{dx}[\ln \Gamma(x)]. \qquad (2.12)$$

Для $\psi(x)$ известны (см., например, [4]) следующие асимптотическое азложение и рекуррентная формула:

$$\psi(x) = \ln x - 1/(2x) - 1/(12x^2) + \ldots; \qquad (2.13)$$

$$\psi(x+1) = \psi(x) + 1/x. \qquad (2.14)$$

Полигамма-функции

$$\psi^{(n)}(x) \equiv d^n\psi(x)/dx^n. \qquad (2.15)$$

$\psi^{(1)}$ называется тригамма-функцией [4]; $\psi^{(2)}$ — тетрагамма-функцией, $\psi^{(3)}$ — пентагамма-функцией и т. д.

Для указанных функций справедливы [4] следующие асимптотичекое разложение и рекуррентная формула:

$$\psi^{(n)}(x) = (-1)^{n+1}\frac{(n-1)!}{x^n}\left(1 + \frac{n}{2x} + \frac{n(n+1)}{12x^2} + \ldots\right); \qquad (2.16)$$

$$\psi^{(n)}(x) = \psi^{(n)}(x+1) + (-1)^{n+1}\, n!/x^{n+1}. \qquad (2.17)$$

МЕТОД РАСЧЕТА

Наличие простых рекуррентных соотношений, которые справедливы во всей области изменения аргумента, а также большая разрядная етка калькулятора позволяют применить следующий способ. Аргуент функций увеличивается на целое положительное число до значения, при котором приведенные усеченные асимптотические разложения ают достаточно малую погрешность. По указанным формулам вычисляются значения функций. Затем производится возвращение к заданым аргументам с помощью рекуррентных формул.

Для Г-функции аргумент увеличивается до 64 + $M(x)$, где M
— дробная часть аргумента. Для дигамма-функции аргумент увели
вается до 20 + $M(x)$. Для полигамма-функций аргумент достато
увеличить до 10 + $M(x)$.

ПРОГРАММА 8

ГАММА-ФУНКЦИЯ Г (x) $(x \leqslant 69)$

	0	1	2	3	4	5
0	$\boxed{\text{С/П} \; 78}$	F5 52	1 14	2 24	× 26	F, 45
1	1 14	+ 96	P6 61	F5 52	1 14	P÷ 33
2	÷ 36	P7 71	F/−/ 55	F/−/ 55	F/−/ 55	F/−/ 55
3	F/−/ 55	F/−/ 55	P8 81	F4 42	↑ 06	F7 72
4	x^y 38	P7 71	P× 23	2 24	× 26	↑ 06
5	F5 52	÷ 36	FВП 65	↑ 06	F6 62	× 26
6	↑ 06	F7 72	× 26	↑ 06	F8 82	× 26
7	P8 81	F5 52	1 14	− 86	P6 61	F8 82
8	↑ 06	F6 62	÷ 36	P8 81	F6 62	1 14
9	− 86	P6 61	F3 32	− 86	PПП 69	FC_x 75

Регистры памяти	2	3	4	5	6	7	8
		$x+1$	$M(x)$	$64+M(x)$			Г $(x$

Порядок работы

а) $x < 64$

1. *Ввод* $(x+1)$ в <3>.

2. *Ввод* $M(x)$ — дробной части аргумента x в <4>. $M(x)$
при $x > 0$; $M(x) = 0$ при x целом; $M(x) < 0$ при $x < 0$.

3. *Ввод* $64 + M(x)$ в <5>.

4. *Пуск* В/О С/П .

5. *Результат*: Г (x) содержится в <8>.

б) $64 \leqslant x \leqslant 69$. В отличие от пп. 2, 3 *следует ввести* $(x +$
<5> и $(x - 63)$ в <4>. Остальные действия такие же, как и
$x < 64$.

При $x > 69$ Г $(x) > 10^{100}$ и возникает переполнение разрядной
кн.

Относительная погрешность меньше $5 \cdot 10^{-5}$.

Время счета $t \approx 2{,}5\ (1 - x/70)$ мин при $x < 64$. При $x \geqslant 64$ $t <$ < 1 мин. Время счета максимально для больших отрицательных x. бращаем внимание читателя на то, что при целых $x < 0$ $\Gamma(x)$ имеет олюса.

Примеры:

$\Gamma(1{,}395) = 0{,}8875524$. Табличное значение 0,8875547;
$\Gamma(68{,}5) = 3{,}001968 \cdot 10^{95}$. Табличное значение $3{,}0019615 \cdot 10^{95}$;
$\Gamma(-3{,}5) = 0{,}2700902$. Табличное значение 0,2700881.

Для последнего примера исходные данные следующим образом располаются по регистрам памяти:

$-2{,}5$ в <3>, $-0{,}5$ в <4>, 63,5 в <5>.

ПРОГРАММА 9

ЛОГАРИФМИЧЕСКАЯ ПРОИЗВОДНАЯ ГАММА-ФУНКЦИЯ $\psi(x)$

	0	1	2	3	4	5
0	$\boxed{}$	F3 32	2 24	× 26	F, 45	P5 51
1	F/–/ 55	3 34	÷ 36	/–/ 56	↑ 06	F5 52
2	– 86	↑ 06	F3 32	P XY 13	+ 96	P5 51
3	F3 32	1 14	– 86	P3 31	F, 45	/–/ 56
4	↑ 06	F5 52	+ 96	P5 51	F2 22	↑ 06
5	F3 32	– 86	PB/O 49	× 26	С/П 78	
6						
7						
8						
9						

егистры амяти	2	3	4	5	6	7	8
	x	$20 + M(x)$		$\psi(x)$			

Порядок работы

а) $x < 20$

1. *Ввод* $20 + M(x)$ в <3>. $M(x)$ — дробная часть аргумента x определяется так же, как в предыдущей программе).
2. *Ввод* x в <2>.
3. *Пуск* В/О С/П.
4. *Результат*: $\psi(x)$ содержится в <5>.

б) $x \geqslant 20$. В отличие от пп. 1, 2 в <2> и в <3> следует вво
$(x+1)$.

Относительная погрешность меньше $2 \cdot 10^{-5}$.

Время счета $t \approx 2\,(1 - x/30)$ мин при $x < 20$. При $x \geqslant 20$ $t < 1$

Примеры.
$\psi(1{,}75) = 0{,}2474730$. Табличное значение 0,24747245... [4];
$\psi(100) = 4{,}600161$. Табличное значение 4,60016185...

ПРОГРАММА 10

ПОЛИГАММА ФУНКЦИИ $\psi^{(n)}(x)$

	0	1	2	3	4	5
0	С/П 78	F4 42	↑ 06	F3 32	÷ 36	P5 51
1	↔xy 16	1 14	+ 96	× 26	6 64	÷ 36
2	↑ 06	F3 32	F/−/ 55	÷ 36	↑ 06	F5 52
3	+ 96	2 24	+ 96	P5 51	F4 42	↑ 06
4	F3 32	x^y 38	× 26	F, 45	↑ 06	F5 52
5	× 26	2 24	÷ 36	P5 51	F3 32	1 14
6	− 86	P3 31	F4 42	1 14	+ 96	↑ 06
7	F3 32	x^y 38	F, 45	↑ 06	F5 52	+ 96
8	P5 51	F2 22	↑ 06	F3 32	− 86	PВ/О 49
9	5 54	F5 52	↑ 06	F6 62	× 26	P5 51

Регистры памяти	2	3	4	5	6	7
	x	$10+M(x)$	n	$\psi^{(n)}(x)$	$(-1)^{n+1}n!$	

Порядок работы

а) $x < 10$

1. Ввод x в <2>.
2. Ввод $10 + M(x)$ в <3>. $M(x)$ — дробная часть аргумен определяется так же, как в программах 8 и 9.
3. *Ввод порядка* полигамма-функции в <4>.
4. *Вычисление* $(-1)^{n+1}\,n!$ и ввод соответствующего числа в <
5. *Пуск* В/О С/П .
6. *Результат*: $\psi^{(n)}(x)$ выдается на индикатор и хранится в <

Относительная погрешность меньше $2 \cdot 10^{-4}$. Для $x < 10$ по ность меньше $5 \cdot 10^{-5}$. Для $x < 2$ погрешность меньше $1 \cdot 10^{-5}$.

б) $x \geqslant 10$. В отличие от пп. 1, 2 в регистры <2> и <3> сл ввести $(x+1)$.

Время счета $t \approx 2(1 - x/20)$ мин при $x < 10$. При $x \geqslant 10$ емя счета $t \approx 1$ мин.

Примеры:
$\psi^{(3)}(1{,}93) = 0{,}561443$. Табличное значение [4] 0,5614423...;
$\psi^{(2)}(1) = -2{,}404114$. Табличное значение $-2{,}4041138$...
Для первого примера исходные данные следующим образом располагаются регистрам памяти:
1,93 в <2>, 10,93 в <3>, 3 в <4>, 3! = 6 в <6>.

2.5. ИНТЕГРАЛ ВЕРОЯТНОСТИ, ИНТЕГРАЛЫ ФРЕНЕЛЯ

ОПРЕДЕЛЕНИЯ

Интеграл вероятности

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt. \tag{2.18}$$

Интегралы Френеля

$$C(x) = \int_0^x \cos\left(\frac{\pi}{2} t^2\right) dt; \tag{2.19}$$

$$S(x) = \int_0^x \sin\left(\frac{\pi}{2} t^2\right) dt. \tag{2.20}$$

Определения (2.19), (2.20) соответствуют [4] и [5]. В [2] и [6] даны сколько иные определения:

$$C(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \frac{\cos t}{\sqrt{t}}\, dt = \sqrt{\frac{2}{\pi}} \int_0^{\sqrt{x}} \cos(t^2)\, dt; \tag{2.21}$$

$$S(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \frac{\sin t}{\sqrt{t}}\, dt = \sqrt{\frac{2}{\pi}} \int_0^{\sqrt{x}} \sin(t^2)\, dt. \tag{2.22}$$

нако таблицы интегралов Френеля чаще соответствуют (2.19) и (2.20).

МЕТОД РАСЧЕТА

Интеграл вероятности легко рассчитывается суммированием ряда

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} e^{-x^2} \sum_{n=0}^{\infty} \frac{2^n x^{2n+1}}{1\cdot 3\cdot \ldots \cdot (2n+1)}. \tag{2.23}$$

Поскольку ряд не является знакочередующимся, указанный метод реализации на микрокалькуляторе применим для весьма больших см. § 2.3). Следует, однако, иметь в виду, что при $x > 3$ значения (x) настолько приближаются к 1, что разность $\operatorname{erfc}(x) \equiv 1 - \operatorname{erf}(x)$, ываемая *дополнительным интегралом вероятности* [4], становит- по порядку величины близкой к ошибкам округления при вычисле-

ак. 1219

нии членов в (2.23). Поэтому (2.23) непригодна для расчета er
при больших x. В этом случае целесообразно использовать асим
ческое разложение для дополнительного интеграла вероятност

$$\operatorname{erfc}(x) \sim \frac{e^{-x^2}}{x\sqrt{\pi}}\left[1 + \sum_{m=1}^{\infty} (-1)^m \frac{1\cdot 3\cdot \ldots \cdot (2m-1)}{(2x^2)^m}\right].$$

В программе 12 учитываются члены до $m = 8$ включительно.
$x > 4$ это дает относительную погрешность менее $1\cdot 10^{-6}$; при $x >$
менее $1\cdot 10^{-5}$; при $x > 3$ — менее $2\cdot 10^{-4}$. Выбранное число ч
$m = 8$, по-видимому, максимально, так как увеличение m практи
не повышает точность при больших x, но снижает при малых x. Отм
что, несмотря на большую относительную погрешность в erfc (
сравнительно малых x, погрешность в erf (x) оказывается еще н
шой. Так, при $x = 2$ относительная погрешность в вычислении
с помощью (2.24) не превышает $4\cdot 10^{-4}$, а при $x = 2{,}5$ не более 1

Интегралы Френеля для $x < 3{,}5$ вычисляются прямым инте
ванием по формуле Симпсона (см. § 2.10). При $x > 3{,}5$ сущес
возрастает время счета, которое пропорционально x^2. Поэтом
$x > 3{,}5$ и даже при меньших x целесообразно использовать асим
ческие разложения [4]:

$$C(x) \sim \frac{1}{\pi x}\left[\frac{1}{y}\left(\frac{15}{y^2} - 1\right)\cos\frac{y}{2} - \left(\frac{3}{y^2} - 1\right)\sin\frac{y}{2}\right] + \ldots;$$

$$S(x) \sim \frac{1}{\pi x}\left[\frac{1}{y}\left(\frac{15}{y^2} - 1\right)\sin\frac{y}{2} - \left(\frac{3}{y^2} - 1\right)\cos\frac{y}{2}\right] + \ldots,$$

где $y = \pi x^2$.

Разложения (2.25) и (2.26) реализованы в программе 14.

ПРОГРАММА 11

ИНТЕГРАЛ ВЕРОЯТНОСТИ erf (x)

	0	1	2	3	4	5
0		F2 22	F/−/ 55	2 24	× 26	P3 31
1	FВП 65	P7 71	P5 51	1 14	P4 41	1 14
2	ВП 66	7 74	/−/ 56	P6 61	F4 42	2 24
3	+ 96	P4 41	↑ 06	F7 72	XY 16	÷ 36
4	↑ 06	F3 32	× 26	P7 71	↑ 06	F5 52
5	+ 96	P5 51	÷ 36	↑ 06	F6 62	− 86
6	РПП 69	2 24	F2 22	F/−/ 55	/−/ 56	P÷ 33
7	2 24	FВП 65	× 26	↑ 06	P× 23	FВП 65
8	÷ 36	↑ 06	F5 52	× 26	P5 51	С/П 78
9						

erf (x)

…нстры …яти	2	3	4	5	6	7	8
	x						

Порядок работы

1. *Ввод* x в <2>.
2. *Пуск* В/О С/П .
3. *Результат*: erf (x) выдается на индикатор и хранится в <5>.

Относительная погрешность меньше $5 \cdot 10^{-6}$.

Время счета $t \approx 2$ мин при $2 \leqslant x \leqslant 3$, $t \approx 1$ мин при $x \leqslant 1$.

Примеры:

erf (0,85) = 0,7706679. Табличное значение 0,77066805;

erf (3) = 0,9999780. Табличное значение 0,9999779.

При $x > 3{,}5$ $[1 - \text{erf}(x)] < 1 \cdot 10^{-6}$. Поэтому использование данной програм… для $x > 4$ не имеет смысла. Следует полагать при $x > 4$ erf (x) = 1 (см. про…мму 12).

ПРОГРАММА 12

ДОПОЛНИТЕЛЬНЫЙ ИНТЕГРАЛ ВЕРОЯТНОСТИ erfc (x)

	0	1	2	3	4	5
0		F2 22	F/−/ 55	F, 45	2 24	÷ 36
1	/−/ 56	P3 31	1 74	P4 41	P5 51	P6 61
2	F4 42	↑ 06	F5 52	× 26	↑ 06	F3 32
3	× 26	P5 51	↑ 06	F6 62	+ 96	P6 61
4	F4 42	2 24	+ 96	P4 41	1 14	7 74
5	− 86	PB/0 49	↔xy 16	F2 22	↑ 06	F/−/ 55
6	P÷ 33	× 26	↑ 06	P× 23	FВП 65	× 26
7	F, 45	↑ 06	F6 62	× 26	P3 31	С/П 78
8						
9						

erfc (x)

…ры	2	3	4	5	6	7	8
	x						

Порядок работы

1. *Ввод* x в <2>.
2. *Пуск* В/О С/П.
3. *Результат*: $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$ выдается на индикатор и хра нится в регистре <3>.

Относительная погрешность меньше: $1\cdot10^{-6}$ при $x > 4$; $1\cdot10^{-}$ при $x > 3{,}6$; $1\cdot10^{-4}$ при $x > 3$; $1\cdot10^{-3}$ при $x > 2{,}6$; $1\cdot10^{-2}$ пр $x > 2{,}3$; $1\cdot10^{-1}$ при $x > 2$.

Время счета примерно 40 с.

Пример.

$\operatorname{erfc}(\sqrt{20}) = \operatorname{erfc}(4{,}4721359) = 2{,}5396322 \cdot 10^{-10}$. Табличное значен $2{,}5396317 \times 10^{-10}$.

ПРОГРАММА 13

ИНТЕГРАЛЫ ФРЕНЕЛЯ $C(x)$, $S(x)$ ($x \leqslant 3{,}5$). ИНТЕГРИРОВАНИЕ ПО ФОРМУЛЕ СИМПСОНА

	0	1	2	3	4	5
0	В/О 48	F7 72	4 44	+ 96	F⊢/ 55	F, 46
1	↑ 06	F6 62	× 26	P4 41	2 24	÷ 36
2	P3 31	↑ 06	F6 62	+ 96	P2 21	ПП 68
3	F7 72	ПП 68	F7 72	+ 96	+ 96	+ 96
4	P5 51	F2 22	↑ 06	F4 42	− 86	РВ/О 49
5	P6 61	ПП 68	F7 72	+ 96	P5 51	БП 58
6	P3 31	F5 52	↑ 06	F3 32	× 26	3 34
7	÷ 36	С/П 78	F2 22	↑ 06	F3 32	− 86
8	P2 21	F/−/ 55	↑ 06	P× 23	× 26	2 24
9	÷ 36	P−(P+) 83(93)	↑ 06	F5 52	+ 96	P5 51

	$S(x)$			0	x	$A(x)$	
Регистры памяти	2	3	4	5	6	7	8
	$C(x)$			1	x	$A(x)$	

Порядок работы

а) Вычисление $C(x)$

1. *Ввод* x в <6>.
2. *Ввод* целой части аргумента $A(x)$ в <7> (округление x п изводить в бóльшую сторону).

3. *Ввод* 1 в <5>.
4. *Пуск* В/О С/П.
5. *Результат*: $C(x)$ выдается на индикатор. В регистре <5> хранится величина $6C(x)[A(x)+4]^2/x$.

б) Вычисление $S(x)$

1. *Замена* в программе команды Р— в ячейке 91 на команду Р+. Соответствующая команда и ее код помещены в скобках в ячейке 91 (см. программу).
2. *Ввод* 0 в <5>.

Остальные операции (ввод x и $A(x)$, пуск и выдача результата) — такие же, как и при вычислении $C(x)$.

Относительная погрешность меньше $5\cdot10^{-5}$.

Время счета $t\approx3(1+x/3)^2$ мин.

Примеры:

C (3,5) = 0,5325683. Табличное значение 0,5325724;
S (3,5) = 0,4152441. Табличное значение 0,4152480.

ПРОГРАММА 14

ИНТЕГРАЛЫ ФРЕНЕЛЯ $C(x)$, $S(x)$ ($x\geqslant3{,}5$). АСИМПТОТИЧЕСКИЕ РАЗЛОЖЕНИЯ (2.25), (2.26)

	0	1	2	3	4	5
0	В/О 48	F2 22	↑ 06	P× 23	× 26	P2 21
1	× 26	P3 31	2 24	÷ 36	P↑ 03	P4 41
2	⇄xy 16	P5 51	F3 32	F, 45	F/—/ 55	3 34
3	× 26	1 14	— 86	P7 71	6 54	× 26
4	4 44	+ 96	↑ 06	F3 32	÷ 36	P8 81
5	F4 42	↑ 06	ПП 68	F7 72	P6 61	F5 52
6	/—/ 56	↑ 06	F4 42	P5 51	ПП 68	F7 72
7	P5 51	С/П 78	F7 72	× 26	P3 31	F8 82
8	↑ 06	F6 52	× 26	↑ 06	F3 32	+ 96
9	↑ 06	F2 22	÷ 36	2 24	F, 45	+ 96

Регистры памяти	2	3	4	5	6	7	8
	x			$C(x)$	$S(x)$		

Порядок работы

1. *Ввод* x в <2>.
2. *Пуск* В/О С/П.
3. *Результаты*: C (x) выдается на индикатор и хранится в рег
<5>; S (x) находится в регистре <6>.

Относительная погрешность меньше: $1\cdot10^{-5}$ при $x > 3{,}5$; 3 · при $x > 3$; $1\cdot10^{-3}$ при $x > 2$.

Время счета примерно 20 с.

Примеры:

C (2,1) = 0,5815215. Табличное значение 0,5815641;
S (2,1) = 0,3746257. Табличное значение 0,3742734;
C (3,52) = 0,5500594. Табличное значение 0,5500611;
S (3,52) = 0,4248708. Табличное значение 0,4248672.

2.6. ФУНКЦИИ БЕССЕЛЯ, МОДИФИЦИРОВАННЫЕ ФУНКЦИИ БЕССЕЛЯ

ОПРЕДЕЛЕНИЯ

Функции Бесселя первого рода $J_\nu(x)$ *и функции Бесселя вт рода (функции Неймана)* $N_\nu(x)$ удовлетворяют дифференциал уравнению

$$\frac{d^2 Z_\nu}{dx^2}+\frac{1}{x}\frac{dZ_\nu}{dx}+\left(1-\frac{\nu^2}{x^2}\right)Z_\nu=0.$$

$J_\nu(x)$ характеризуется следующим представлением в виде

$$J_\nu(x)=\left(\frac{x}{2}\right)^\nu\sum_{k=0}^{\infty}\frac{(-x^2/4)^k}{k!\,\Gamma(\nu+k+1)}.$$

$N_\nu(x)$ при нецелом ν связана с $J_\nu(x)$ соотношением

$$N_\nu(x)=[J_\nu(x)\cos(\nu\pi)-J_{-\nu}(x)]/\sin(\nu\pi).$$

При целых ν величина N_ν получается из (2.28) предельным пере

Модифицированные функции Бесселя $I_\nu(x)$ и $K_\nu(x)$ удовлет дифференциальному уравнению

$$\frac{d^2 Z_\nu}{dx^2}+\frac{1}{x}\frac{dZ_\nu}{dx}-\left(1+\frac{\nu^2}{x^2}\right)Z_\nu=0.$$

$I_\nu(x)$ характеризуются рядом

$$I_\nu(x)=\left(\frac{x}{2}\right)^\nu\sum_{k=0}^{\infty}\frac{(x^2/4)^k}{k!\,\Gamma(\nu+k+1)}.$$

$K_\nu(x)$ при нецелом ν связана с $I_\nu(x)$ соотношением, анало (2.28):

$$K_\nu(x)=[\pi(I_{-\nu}(x)-I_\nu(x))]/[2\sin(\nu\pi)].$$

Если ν — целое или нуль, то правая часть этого уравнения за ся его предельным значением.

Сферические функции Бесселя и модифицированные сферические функ-и Бесселя пропорциональны соответствующим функциям Бесселя ррядка $(n + 1/2)$ [4]. Так, сферические функции Бесселя первого и орого рода определяются соотношениями соответственно:

$$j_n(x) = \sqrt{\pi/(2x)}\, J_{n+1/2}(x),$$
$$y_n(x) = \sqrt{\pi/(2x)}\, N_{n+1/2}(x). \qquad (2.31)$$

Аналогично определяются и модифицированные сферические функ-ии Бесселя [4].

МЕТОДЫ РАСЧЕТА

Наиболее простым для реализации на микрокалькуляторе является тод рядов, связанный с использованием формул (2.27) и (2.29). ля функций Бесселя первого рода ряд (2.27) при положительных ν ляется знакочередующимся, что затрудняет его использование и больших x (см. расчет Ei (x) на с. 25). Для $\nu \geqslant 0$ метод рядов жно использовать до $x \leqslant 9$ с погрешностью не более $2 \cdot 10^{-5}$. и увеличении ν и при отрицательных ν область допустимых x рас-ряется. Так, для $\nu = 100$ формула (2.27) применима до $x = 50$. программу 15).

Для модифицированных функций Бесселя I_ν метод рядов (формула 28)) применим для любых ν и $x \geqslant 0$ (кроме целых $\nu < 0$).

Функции Бесселя второго рода и модифицированные функции Бес-ля $K_\nu(x)$ вычисляются соответственно по формулам (2.28) и (2.30) только для дробных $\nu > 0$.

Функции Бесселя первого рода и сферические функции Бесселя рвого и второго рода при произвольных значениях аргумента и по-дка вычисляются с помощью асимптотических разложений Дебая ., например, [4]) и рекуррентных формул для функций Бесселя. зложение Дебая при учете только членов порядка ν^{-1} имеет вид

$$J_\nu(x) = \frac{e^{\nu(\operatorname{th}\alpha - \alpha)}}{2\pi\nu \operatorname{th}\alpha}\left(1 + \frac{3\operatorname{cth}\alpha - 5\operatorname{cth}^2\alpha}{\nu}\right), \qquad (2.32)$$

$\operatorname{ch}\alpha = \nu/x$.

Влияние отброшенных слагаемых более высокого порядка по ν^{-1} ло при $\nu \geqslant 50$ и $x/\nu < 0{,}5$. В этом случае основная погрешность зана с экспоненциальным множителем $e^{\nu(\operatorname{th}\alpha - \alpha)}$. Реальная точ-ть, достижимая на микрокалькуляторе при использовании (2.32) казанных выше значениях ν и x/ν, составляет $\approx 1 \cdot 10^{-5}$.

Переход к $J_\nu(x)$ с малыми значениями порядка ν производится по уррентным формулам

$$Z_{\nu-1} = 2\nu Z_\nu/x - Z_{\nu+1}, \qquad (2.33)$$

Z_ν означает J_ν или N_ν. Эти соотношения образуют (применительно , но не к N_ν!) устойчивую систему разностных уравнений 4, с. 11]) *при убывании* ν. Поэтому многократное применение (2.33) оцессе спуска по ν не приводит к накоплению погрешностей. Ре-нтные формулы с понижением ν могут применяться до отрица-ых ν включительно.

Реальная относительная погрешность при совместном исполь-
нии (2.32) и (2.33) (исключая окрестность нулей $J_\nu(x)$ — см. опис
к программам 15, 17), не хуже нескольких единиц 10^{-5} для произв
ных ν и x.

Описанные методы непригодны для вычисления функций Бес
второго рода и модифицированных функций K_n *целого порядка.*
упомянутых функций рекуррентные формулы образуют устойчи
систему разностных уравнений *при повышении* n (или в общем ν),
позволяет, располагая величинами $N_{\nu_0}(x)$ и $N_{\nu_0+1}(x)$ или K
и $K_{\nu_0+1}(x)$ при малых ν_0, вычислять функции N_{ν_0+m} или соответс
но K_{ν_0+m} при сколь угодно больших m. Далее приведены соотв
вующие программы для N_ν и K_ν. В программе 18 используются ре
рентные формулы (2.33), в программе 20 — аналогичные форму

$$K_{\nu+1}(x) = K_{\nu-1}(x) + 2\nu K_\nu(x)/x.$$

ПРОГРАММА 15

ФУНКЦИИ БЕССЕЛЯ ПЕРВОГО И ВТОРОГО РОДА $J_\nu(x)$, $N_\nu(x)$. МЕТОД РЯДОВ. ФОРМУЛА (2.27) И СООТНОШЕНИЕ (2.28)

	0	1	2	3	4	5
0	$\boxed{\text{С/П}\ 78}$	1 14	Р5 51	Р7 71	F4 42	F/–/ 55
1	/–/ 56	Р2 21	F8 82	1 14	+ 96	↑ 06
2	Р8 81	F3 32	+ 96	× 26	↑ 06	F2 22
3	$\overleftarrow{xy}\rightarrow$ 16	÷ 36	↑ 06	F7 72	× 26	Р7 71
4	↑ 06	F5 52	$\overleftarrow{xy}\rightarrow$ 16	+ 96	Р5 51	− 86
5	РБП 59	F1 12	F6 62	÷ 36	Р5 51	F5 52
6	↑ 06	F3 32	÷ 36	↑ 06	F4 42	× 26
7	Р5 51	Р/–/ 53	F3 32	1 14	− 86	Р3 31
8	/–/ 56	РВ/0 49	F/–/ 55	С/П 78	F6 62	× 26
9	↑ 06	F5 52	− 86	↑ 06	F7 72	÷ 36

Пуск 2:	$-\nu$	$x/2$			$\Gamma(-\nu)\left(\frac{x}{2}\right)^{1+\nu}$	
Регистры памяти	2	3	4	5	6	7
Пуск 1:	ν	$x/2$			$\Gamma(M)\left(\frac{x}{2}\right)^{1-M}$	

Порядок работы при вычислении $J_\nu(x)$

1. *Ввод* ν в $<3>$.
2. *Ввод* $(x/2)$ в $<4>$.
3. *Ввод* 0 в $<8>$.
4. *Ввод* в $<6>$ одной из следующих величин:

   1, если ν — целое положительное или нуль;

   $\Gamma(M)(x/2)^{1-M}$, если ν — дробное и положительное;

   $\Gamma(\nu)(x/2)^{1-\nu}$, если ν — дробное и отрицательное.

   (Γ — гамма-функция, M — дробная часть ν).
5. *Пуск* В/О С/П.
6. *Результат*: В $<5>$ содержится $J_\nu(x)$.

Внимание: При ν, целом или равном нулю, останов происходит по команде переполнения (деление на нуль). Это не отражается на результатах.

Погрешности:

а) $x \leqslant \nu$. При $x \leqslant 7$ относительная погрешность менее $1 \cdot 10^{-8}$ (единиц в 6-й значащей цифре). При $x \leqslant 10$ относительная погрешность меньше $2 \cdot 10^{-4}$. При больших ν диапазон допустимых значений x расширяется. Для $\nu \geqslant 4$ относительная погрешность не превышает $1 \cdot 10^{-4}$ вплоть до

$$x_{\max} \approx 2 + 4\sqrt{\nu}. \tag{2.35}$$

б) $x > \nu$. В этой области функции Бесселя осциллируют и проходят через нули, в окрестности которых указанные оценки несправедливы. Вместо них можно оперировать абсолютными погрешностями. Это имеет определенный смысл, поскольку в области $x > \nu$ значение $J_\nu(x)$ в среднем имеют порядок 0,1. В результате можно рекомендовать следующие критерии. Если вычисленное значение функции *по модулю меньше* 0,1, то абсолютная погрешность: при $x \leqslant 7$ меньше $1 \cdot 10^{-6}$; при $x \leqslant 9$ меньше $2 \cdot 10^{-6}$; при $\nu \geqslant 4$ и $x \leqslant x_{\max}$ меньше $2 \cdot 10^{-5}$. $x_{\max}$ определен в (2.35).

Если вычисленные значения функции по модулю больше или равны 0,1, то справедливы оценки п. а). Время счета растет с увеличением x и падает с ростом ν. Характерные времена видны из приводимых примеров.

Примеры:

$J_0(1{,}7) = 0{,}39798486$. Табличное значение 0,39798485 (1 мин);
$J_0(9) = -0{,}09033399$. Табличное значение $-0{,}09033361$ (2 мин);
$J_{20}(20) = 0{,}16473092$. Табличное значение 0,164742 (3 мин);
$J_{100}(50) = 1{,}116853 \cdot 10^{-21}$. Табличное значение $1{,}11593 \cdot 10^{-21}$ (8 мин);
$J_{-7{,}5}(10) = 0{,}1072490$. Табличное значение 0,1072491 (2 мин).

При вычислениях $J_\nu(x)$ с дробным порядком требуется исполь вать значения гамма-функции. Эту величину легко рассчитать при п мощи программы 8. В табл. 2 приведены значения $\Gamma(x)$ для некотор употребительных величин x.

Таблица 2

x	1/6	1/5	1/4	1/3	2/5	1/2	3/5
$\Gamma(x)$	5,566314	4,590843	3,62561	2,678939	2,218159	1,772454	1,4891
x	2/3	3/4	4/5	—1/5	—1/4	—1/3	—1/2
$\Gamma(x)$	1,354118	1,225417	1,164229	—5,821145	—4,901668	—4,062354	—3,5449

Порядок работы при вычислении $N_\nu(x)$

Расчет выполняется в результате *трех последовательных пу* калькулятора при использовании *одной и той же программы*

1. *Ввод* ν в <3>.
2. *Ввод* $x/2$ в <4>.
3. *Ввод* 0 в <8>.
4. *Ввод* $\Gamma(M)\,(x/2)^{1-M}$ в <6>, где Γ — гамма-функция, M дробная часть ν. При вычислении $N_\nu(x)$ допускаются только пол тельные дробные ν.
5. *Пуск 1*: В/О С/П. После останова в регистре <5> оказ ется $J_\nu(x)$.
6. *Ввод* $(-\nu)$ в <3> и 0 в <8>.
7. *Ввод* $\Gamma(-\nu)\,(x/2)^{1+\nu}$ в <6>.

Содержимое остальных регистров памяти и стека не должно нятъся. Под табличкой регистров памяти показано состояние регис в которые вводятся исходные данные перед пуском 1, а над табл — перед пуском 2.

8. *Пуск 2*: В/О С/П . После останова в регистре <5 функция $J_{-\nu}(x)$.
9. *Ввод* $J_\nu(x)$ в <6>; вычисление $\sin(\nu\pi)$ и $\cos(\nu\pi)$ и их соответственно в регистры <7> и <Y>. Эти операции выполн последовательным нажатием следующих клавиш (9 операций): Р Р6 Р× $\hat{\nu}$ × Р↑ $\overleftarrow{\underrightarrow{xy}}$ Р7 . Здесь $\hat{\nu}$ означает набор числ клавиатуре.

10. *Пуск 3*: С/П .

11. *Результат*: После останова на индикаторе $N_\nu(x)$.

Погрешности такие же, как при вычислении $J_\nu(x)$.

Время счета примерно вдвое большее, чем при вычислении $J_\nu(x)$ при тех же ν и x.

Пример. Вычислить $N_{7,5}(10)$.

Рассчитываем вспомогательные величины, используемые в качестве исходных данных перед пусками 1 и 2. Для $\nu = 7{,}5$ $M = 0{,}5$; $\Gamma(M)(x/2)^{1-M} = \Gamma(0{,}5)\cdot 5^{1/2} = 1{,}772454\cdot 5^{1/2} = 3{,}9633274$. Величину $\Gamma(-7{,}5)$ находим по рекуррентным формулам:

$$\Gamma(-1/2) = (-3/2)\cdot(-5/2)\cdot\ldots\cdot(-15/2)\cdot\Gamma(-15/2),$$

$$\Gamma(-7{,}5) = -\frac{2^7}{3\cdot 5\cdot 7\cdot\ldots\cdot 15}\,\Gamma(-1/2) = 2{,}2384931\cdot 10^{-4},$$

$$\Gamma(-7{,}5)\cdot 5^{1+7{,}5} = 195{,}52431.$$

Значения $\Gamma(1/2)$ и $\Gamma(-1/2)$ взяты из таблицы 2.

Переходим к вычислениям.

Пуск 1: $J_{7,5}(10) = 0{,}2860869$. Табличное значение 0,2860884. Время счета 1,5 мин.

Пуск 2: $J_{-7,5}(10) = 0{,}1072490$. Табличное значение 0,1072491. Время счета 1,5 мин.

Пуск 3: $N_{7,5}(10) = 0{,}1072491$. Время счета 10 с. Для случая $\nu = 7{,}5$ N_ν и $J_{-\nu}$ должны совпадать.

МЕТОД РАСЧЕТА ФУНКЦИЙ БЕССЕЛЯ ПЕРВОГО РОДА $J_\nu(x)$, СФЕРИЧЕСКИХ ФУНКЦИЙ БЕССЕЛЯ ПЕРВОГО И ВТОРОГО РОДА $j_n(x)$, $y_n(x)$, ФУНКЦИЙ БЕССЕЛЯ ВТОРОГО РОДА ПОЛУЦЕЛОГО ПОРЯДКА $N_{n+1/2}(x)$. АСИМПТОТИЧЕСКОЕ РАЗЛОЖЕНИЕ ДЕБАЯ (2.32) И РЕКУРРЕНТНАЯ ФОРМУЛА (2.33)

Последовательно используются две программы. *Программа 16* дает вспомогательные функции $\Phi_P(x)$ и $\Phi_{P-1}(x)$, которые пропорциональны функциям Бесселя больших порядков P ($\Phi_P(x) = -\sqrt{2\pi}\,J_P(x)$). Расчет ведется по формуле (2.32). Найденные величины Φ_P и Φ_{P-1} затем используются в качестве исходных в *програме 17*, которая определяет $J_\nu(x)$, а также $N_\nu(x)$ полуцелого порядка и сферические функции Бесселя (см. (2.31)) первого и второго рода $j_n(x)$ и $y_n(x)$.

Необходимость совместного применения обеих программ усложняет выполнение многократных расчетов при *разных* x. В этом случае целесообразно сначала по программе 16 при каждом x рассчитать пары величин Φ_P и Φ_{P-1} (на вычисление каждой пары требуется 40 с), а затем переходить к набору программы 17.

Если многократные расчеты производятся при одном значении x и *разных* ν, то программа 16 используется однократно, а все нужные варианты рассчитываются далее по программе 17.

ПРОГРАММА 16

ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ $\Phi_P(x)$ и $\Phi_{P-1}(x)$

	0	1	2	3	4	5
0	$\boxed{\text{С/П 78}}$	F7 72	↑ 06	F3 32	÷ 36	P6 61
1	F/—/ 55	1 14	— 86	FВП 65	↑ 06	F6 62
2	+ 96	P$\overleftarrow{xy}$→ 13	P8 81	F6 62	÷ 36	↑ 06
3	P2 21	F8 82	— 86	↑ 06	F7 72	× 26
4	P÷ 33	P8 81	F2 22	F, 45	F/—/ 55	5 54
5	× 26	3 34	— 86	↑ 06	F2 22	÷ 36
6	2 24	4 44	÷ 36	↑ 06	F7 72	÷ 36
7	1 14	— 86	↑ 06	F8 82	× 26	P8 81
8	F7 72	1 14	— 86	P7 71	F2 22	× 26
9	FВП 65	F, 45	↑ 06	F8 82	× 26	P4 41

Регистры памяти	2	3	4	5	6	7	8
		x	$\Phi_{P-1}(x)$	$\Phi_P(x)$		P	

Порядок работы

1. *Ввод* аргумента x функции Бесселя в <3>.
2. *Ввод асимптотического порядка* P в <7>. Величина P дол
быть достаточно большой для обеспечения нужной точности форм
Дебая и отличаться от ν на целое число. Кроме того, должно
$P/x > 2$. Этим условиям удовлетворяют, например, следующие
венства:

$$P = M(\nu) + 50,\ 0 \leqslant x \leqslant 10;$$
$$P = M(\nu) + 100,\ 10 \leqslant x \leqslant 50;$$
$$P = M(\nu) + 2A(x),\ x > 50,$$

где $A(x)$ — целая часть x; $M(\nu)$ — дробная часть ν.

3. *Пуск 1*: В/О С/П . После останова на индикаторе и в
оказывается величина $\Phi_P(x)$. *Эту величину следует заслать в ре*
<5> (*команда* Р5).
4. *Пуск 2*. В/О С/П .
5. *Результаты*: После повторного пуска на индикаторе и в
стре <4> окажется величина $\Phi_{P-1}(x)$, а в регистре <7> —
чина $P - 2$. В регистре <5> после пуска 1 сохраняется Φ_P

При многократных расчетах с различными x записываем получившиеся величины $\Phi_P(x)$, $\Phi_{P-1}(x)$ и P для последующего использования. Далее повторяем процедуру счета при разных x, начиная с п. 1. Если, однако, необходимы величины $J_\nu(x)$ при одном значении x и разных ν, то, не меняя содержимого регистров памяти, переходим к набору программы 17.

Время счета, включая повторный пуск, ≈ 40 с.

Примеры:

$$P=50;\ x=1 \left\{ \begin{array}{l} \Phi_{50}(1) = -7{,}284993\cdot 10^{-80}; \\ \Phi_{49}(1) = -7{,}284128\cdot 10^{-78}; \end{array} \right.$$

$$P=200;\ x=100 \left\{ \begin{array}{l} \Phi_{200}(100) = -5{,}163364\cdot 10^{-41}; \\ \Phi_{199}(100) = -1{,}926907\cdot 10^{-40}; \end{array} \right.$$

$$P=52{,}75;\ x=3 \left\{ \begin{array}{l} \Phi_{52,75}(3) = -2{,}955469\cdot 10^{-60}; \\ \Phi_{51,75}(3) = -1{,}038541\cdot 10^{-58}; \end{array} \right.$$

$$P=50{,}5;\ x=10 \left\{ \begin{array}{l} \Phi_{50,5}(10) = -1{,}410806\cdot 10^{-30}; \\ \Phi_{49,5}(10) = -1{,}411255\cdot 10^{-29}. \end{array} \right.$$

Напомним, что величины $\Phi_P(x)$ пропорциональны $J_P(x)$. Так, $J_{50}(1) = -\Phi_{50}(1)/\sqrt{2\pi} = 2{,}906292\cdot 10^{-80}$. Табличное значение $J_{50}(1)$ $2{,}9060049 \times 10^{-80}$.

ПРОГРАММА 17

ФУНКЦИИ $J_\nu(x)$, $N_{n+1/2}(x)$, $j_n(x)$, $y_n(x)$

	0	1	2	3	4	5
0	$\boxed{\text{В/О 48}}$	ПП 68	/–/ 56	F5 62	/–/ 56	Р8 81
1	F2 22	/–/ 56	Р2 21	ПП 68	/–/ 56	Р× 23
2	2 24	× 26	FВП 65	F, 45	↑ 06	F8 82
3	× 26	Р7 71	F5 52	× 26	Р4 41	F3 32
4	4 44	× 26	FВП 65	F, 45	↑ 06	F8 82
5	× 26	Р8 81	F5 52	× 26	Р5 51	С/П 78
6	F7 72	1 14	— 86	Р7 71	F3 32	÷ 36
7	2 24	× 26	↑ 06	F4 42	Р6 61	× 26
8	↑ 06	F5 52	— 86	Р4 41	F6 62	Р5 51
9	F7 72	↑ 06	F2 22	— 86	РПП 69	/–/ 56

			$(-1)^n N_{n+1/2}(x)$	$(-1)^n y_n(x)$		$J_\nu(x)$	$j_n(x)$
Регистры памяти	2	3	4	5	6	7	8
	ν	x	$\Phi_{P-1}(x)$	$\Phi_P(x)$		$P-1$	

Порядок работы

1. *Ввод* x в <3>.
2. *Ввод* $\Phi_{P-1}(x)$, $\Phi_P(x)$ в <4> <5> соответственно.
3. *Ввод* $P-1$ в <7>.
4. *Ввод требуемого* ν в <2>. С этой величиной и с x должно бы согласовано значение P (условия (2.36)). Обращаем внимание на т что величины P в программах 16 и 17 должны совпадать.

При однократных расчетах пп. 1 и 2 можно опустить, если пос окончания работы программы 16 не изменялось содержимое регистр <3>, <4> и <5>.

5. *Пуск* В/О С/П .
6. *Результаты*:

Регистр <7> $J_\nu(x)$. ν и $x \geqslant 0$ любые;
регистр <8> $j_n(x)$, когда $\nu = n + 1/2$ ($n > 0$ и целое);
регистр <4> $(-1)^n N_{n+1/2}(x)$, когда $\nu = n + 1/2$ ($n > 0$ и цело
регистр <5> $(-1)^n y_n(x)$, когда $\nu = n + 1/2$ ($n > 0$ и цело

7. *Повторение вычисления с другими* ν *и теми же* x производит путем ввода прежних значений $\Phi_P(x)$, $\Phi_{P-1}(x)$ и $(P-1)$ в регист <4>, <5>, <7> соответственно и нового значения ν в <2 а затем — команда пуска В/О С/П .

Время счета $t \approx (P + \nu)/15$ мин.

Погрешности в результате совместного использования програ 16 и 17: при $x \leqslant \nu$ относительная погрешность примерно $1 \cdot 10^{-7}$. П $x > \nu$ эта оценка сохраняется за исключением окрестности нулей фу ций. Если вычисленное значение функций $J_\nu(x)$ и $N_\nu(x)$ при x по модулю меньше 0,1, то *абсолютная* погрешность менее $3 \cdot 10^{-8}$ подробнее комментарий к программе 15). *Относительная* погрешн *сферических функций Бесселя* совпадает с аналогичной погрешнос соответствующих функций Бесселя первого и второго рода. Поэт для нахождения погрешности функций $j_n(x)$ и $y_n(x)$ в области x нужно оценить *относительные* погрешности соответствующих фун Бесселя по указанной *абсолютной* погрешности и их вычислен значениям.

Примеры:

1. $J_{0,75}(3) = 0,2162035$. Табличное значение 0,2162 (см. [6]).
2. $J_{2,5}(3) = 0,4127205$. Табличное значение 0,4127;
$N_{2,5}(3) = -0,3690499$;
$j_2(3) = 0,2986450$. Табличное значение [4] 0,29863750;
$y_2(3) = -0,2670449$. Табличное значение $-0,26703834$ (вычи по $P = 52,5$).
3. $J_{7,5}(10) = 0,2861203$;
$j_7(10) = 0,1133988$. Табличное значение 0,1133862;
$N_{7,5}(10) = 0,1072611$;
$y_7(10) = 0,04251110$. Табличное значение 0,04250633.

ПРОГРАММА 18

ФУНКЦИИ БЕССЕЛЯ ВТОРОГО РОДА $N_\nu\ (x)$. РЕКУРРЕНТНЫЕ ФОРМУЛЫ (2.33)

Предыдущая программа неприменима для расчета функций $N_\nu\ (x)$ целого порядка. Если известны функции $N_0\ (x)$ и $N_1\ (x)$ или в общем $N_{\nu_0}\ (x)$ и $N_{\nu_0-1}\ (x)$ с малыми ν_0, то функции того же аргумента порядка $(\nu_0 + m)$, где m — целое, могут быть вычислены с помощью рекуррентных формул, которые образуют устойчивую разностную систему *при увеличении* m.

	0	1	2	3	4	5
0		F4 42	1 14	+ 96	P4 41	F2 22
1	÷ 36	2 24	× 26	↑ 06	F5 52	P7 71
2	× 26	↑ 06	F6 62	− 86	P5 51	F7 72
3	P6 61	F3 32	1 14	− 86	↑ 06	F4 42
4	− 86	РПП 69	P0 01	С/П 78		
5						
6						
7						
8						
9						

				$N_\nu(x)$	$N_{\nu-1}(x)$		
Регистры памяти	2	3	4	5	6	7	8
	x	ν	ν_0	$N_{\nu_0}(x)$	$N_{\nu_0-1}(x)$		

Порядок работы

1. *Ввод аргумента* x в <2>.
2. *Ввод порядка искомой функции* ν в <3>.
3. *Ввод исходного порядка* ν_0 в <4>. ν_0 должен отличаться от ν на целое число.
4. *Ввод значений исходных функций* $N_{\nu_0}\ (x)$ и $N_{\nu_0-1}\ (x)$ соответственно в <5> и <6>.
5. *Пуск* В/О С/П .
6. *Результаты*: N_ν — в регистре <5>, $N_{\nu-1}$ — в регистре <6>. Если требуется дополнительно рассчитать N_{ν_1}, где $\nu_1 > \nu$, то до-

статочно, не меняя содержимого остальных регистров, ввести ν_1 в <3> и осуществить пуск В/О С/П .

Время счета $t \approx (\nu - \nu_0)/15$ мин.

Погрешность определяется точностью задания исходных функций $N_{\nu_0}(x)$, $N_{\nu_0-1}(x)$. В области $x > \nu$ при вычисленных значениях функций, меньших по модулю 0,1, следует принимать во внимание только абсолютную погрешность, ввиду наличия в этой области нулей (см. подробнее комментарий к программе 15). Здесь абсолютная погрешность (при достаточной точности исходных N_{ν_0} и N_{ν_0-1}) меньше $1 \cdot 10^{-7}$. В остальных случаях предельная относительная погрешность $1 \cdot 10^{-6}$.

Пример. Вычислить $N_{50}(5)$, $N_{49}(5)$, если известны [4]:
$N_0(5) = -3{,}0851763 \cdot 10^{-1}$, $N_1(5) = 1{,}4786314 \cdot 10^{-1}$.

Результаты:
$N_{50}(5) = -2{,}788834 \cdot 10^{42}$. Табличное значение $-2{,}7888370 \cdot 10^{42}$;
$N_{49}(5) = -1{,}426676 \cdot 10^{11}$.

Время счета примерно 3,5 мин.

ПРОГРАММА 19

МОДИФИЦИРОВАННЫЕ ФУНКЦИИ БЕССЕЛЯ $I_\nu(x)$ и $K_\nu(x)$. МЕТОД РЯДОВ. ФОРМУЛА (2.29) И СООТНОШЕНИЕ (2.30)

	0	1	2	3	4	5
0	С/П 78	Р8 81	1 14	Р7 71	Р5 51	F4 42
1	F/−/ 65	Р2 21	F8 82	1 14	+ 96	↑ 06
2	Р8 81	F3 32	+ 96	× 26	↑ 06	F2 22
3	xy 16	÷ 36	↑ 06	F7 72	× 26	Р7 71
4	↑ 06	F5 52	xy 16	+ 96	Р5 51	− 86
5	РБП 59	F1 12	F6 62	÷ 36	Р5 51	F5 52
6	↑ 06	F3 32	÷ 36	↑ 06	F4 42	× 26
7	Р5 51	Р/−/ 53	F3 32	1 14	− 86	Р3 31
8	/−/ 56	РВ/0 49	F/−/ 55	С/П 78	F5 52	↑ 06
9	Р, 43	Р, 43	− 86	2 24	÷ 36	Р6 61

Пуск 2:		$-\nu$	$x/2$		$\Gamma(-\nu)\left(\frac{x}{2}\right)^{1+\nu}$	
Регистры памяти	2	3	4	5	6	7
Пуск 1:		ν	$x/2$		$\Gamma(M)\left(\frac{x}{2}\right)^{1-M}$	

Порядок работы

Для полного расчета обеих функций предполагается три последовательных пуска микрокалькулятора с промежуточным вводом дополнительных данных (аналогично программе 15). Величина $I_\nu(x)$ находится после первого пуска. Для получения $K_\nu(x)$ требуется произвести два дополнительных пуска.

1. *Ввод* ν в <3>.
2. *Ввод* $(x/2)$ в <4>.
3. *Ввод* в регистр <6> одной из следующих величин: 1, если ν — целое и положительное; $\Gamma(M)(x/2)^{1-M}$, если ν — дробное и положительное; $\Gamma(\nu)(x/2)^{1-\nu}$, если ν — дробное и отрицательное (Γ — гамма-функция; M — дробная часть ν; значения $\Gamma(x)$ для некоторых x приведены в комментариях к программе 15).
4. *Пуск 1.* В/О С/П . *Перед пуском на индикаторе должен быть 0.* После останова в регистре <5> оказывается величина $I_\nu(x)$. Дальнейшие операции связаны с вычислением $K_\nu(x)$.
5. *Ввод* $(-\nu)$ в <3>.
6. *Ввод* $\Gamma(-\nu)(x/2)^{1+\nu}$ в <6>.

Состояние остальных регистров памяти и стека после пуска 1 не должно изменяться. В табличке регистров памяти снизу показано содержимое регистров памяти после ввода исходных данных перед пуском 1, а сверху — перед пуском 2.

7. *Пуск 2.* В/О С/П . Перед пуском на *индикаторе должен быть 0.* После останова в регистре <5> — функция $I_{-\nu}(x)$.
8. *Пуск 3.* С/П . После останова на индикаторе и в регистре <6> функция $\sin(\nu\pi) K_\nu(x)/\pi$.
9. *Домножение предыдущего результата* на $\pi/\sin(\nu\pi)$. Эта операция выполняется в результате последовательного нажатия на следующие клавиши P× ν × P+ ÷ ↑ F6 × . Здесь ν означает набор числа ν на клавиатуре. После исполнения указанных операций на индикаторе оказывается $K_\nu(x)$.

Погрешности. Относительная погрешность вычисления $I_\nu(x)$ меньше $1\cdot10^{-5}$ при всех допустимых значениях ν и x (исключаются $x<0$ и целые отрицательные ν). Для $K_\nu(x)$ алгоритм, связанный с использованием формулы (2.30), применим лишь при $x \leqslant 5$, так как для больших x величины $I_\nu(x)$ и $I_{-\nu}(x)$ практически совпадают, тогда как их разность определяет $K_\nu(x)$. В результате за счет погрешности округления погрешность быстро нарастает. При $x = 5$ относительная погрешность $K_\nu(x) \approx 0{,}01$. При $x = 3$ относительная погрешность менее $\cdot 10^{-1}$.

Время счета растет с увеличением x (см. примеры)

Примеры:

$I_2(3{,}4) = 3{,}4494587$. Табличное значение [4] 3,4494589 (1 мин);

$I_{-0,5}(10) = 2778{,}7843$. Табличное значение [7] 2778,7846 (1 мин);

$I_2(50) = 2{,}8164228\cdot10^{20}$. Табличное значение $2{,}816431\cdot10^{20}$ (3,5 мин);

$K_{1/3}(5) = 0{,}003763633$. Табличное значение 0,003728913 (2 мин).

После пуска 1 в регистре <5> $I_{1/3}(5) = 26{,}897577$;

после пуска 2 в регистре <5> $I_{-1/3}(5) = 26{,}899652$;
после пуска 3 в регистре <6> $1{,}0375 \cdot 10^{-3}$);

$K_{1/3}(3) = 0{,}03531322$. Табличное значение 0,03530521 (2 мин);

$K_{1/3}(1{,}6) = 0{,}1932450$. Табличное значение 0,19324 (2 мин).

Обращаем внимание читателя на то, что для получения макси мальной точности исходные данные желательно вычислять на микро калькуляторе с засылкой промежуточных значений величин в памят калькулятора. При этом удерживается восьмая значащая цифра, кот рая при индикации смешанных чисел не высвечивается на индикатор (см. § 1.1).

ПРОГРАММА 20

МОДИФИЦИРОВАННЫЕ ФУНКЦИИ БЕССЕЛЯ $K_\nu(x)$. РЕКУРРЕНТНЫЕ ФОРМУЛЫ (2.34)

Указанные формулы образуют устойчивую разностную систе *при увеличении порядка* ν. Предполагаются известными функц $K_{\nu_0}(x)$ и $K_{\nu_0-1}(x)$. Вычисляются функции $K_{\nu_0+m}(x)$, где m — про вольное целое положительное число. ν_0 может быть любым.

	0	1	2	3	4	5
0		F4 42	1 14	+ 96	Р4 41	F2 22
1	÷ 36	2 24	× 26	↑ 0б	F5 52	Р7 71
2	× 26	↑ 0б	F6 62	+ 96	Р5 51	F7 72
3	Р6 61	F3 32	1 14	− 86	↑ 0б	F4 42
4	− 86	РПП 69	Р0 01	С/П 78		
5						
6						
7						
8						
9						

				K_ν	$K_{\nu-1}$	
Регистры памяти	2	3	4	5	6	7
	x	ν	ν_0	K_{ν_0}	K_{ν_0-1}	

Порядок работы

1. *Ввод аргумента* x в $<2>$.
2. *Ввод порядка искомой функции* ν в $<3>$.
3. *Ввод начального порядка* ν_0 в $<4>$. ν_0 должен отличаться от ν на целое число.
4. *Ввод значений* исходных функций $K_{\nu_0}(x)$ и $K_{\nu_0-1}(x)$ соответст-ветственно в $<5>$ и $<6>$.
5. *Пуск* В/О С/П .
6. *Результаты*: K_ν в регистре $<5>$; $K_{\nu-1}$ — в регистре $<6>$. Если требуется дополнительно рассчитать $K_{\nu_1}(x)$, где разность $(\nu_1-\nu)$ является целым положительным числом, то достаточно, не меняя содержимого остальных регистров, ввести ν_1 в $<3>$ и затем нажать клавиши пуска В/О С/П.

Время счета $t \approx (\nu-\nu_0)/15$ мин.

Погрешность определяется точностью задания исходных функций K_{ν_0} и K_{ν_0-1}. При максимальной точности этих функций, допускаемой микрокалькулятором, относительная погрешность меньше $1\cdot10^{-6}$.

Пример. Вычислить $K_{20}(17{,}4)$, если даны [4]:
$K_0(17{,}4) = 8{,}279921\cdot10^{-9}$; $K_1(17{,}4) = 8{,}514612\cdot10^{-9}$.
Результат: $K_{20}(17{,}4) = 2{,}588878\cdot10^{-4}$. Табличное значение $2{,}5889\cdot10^{-4}$.

2.7. ФУНКЦИИ ЭЙРИ

ОПРЕДЕЛЕНИЯ

Функции Эйри удовлетворяют дифференциальному уравнению $y''-xy=0$. Функции Ai (x) и Bi (x) — пара линейно независимых решений этого уравнения. Для них имеют место следующие разложения в степенные ряды [4]:

$$\mathrm{Ai}(x) = C_1 f(x) - C_2 g(x); \tag{2.37}$$

$$\mathrm{Bi}(x) = \sqrt{3}\,[C_1 f(x) + C_2 g(x)], \tag{2.38}$$

где

$$f(x) = 1 + \frac{1}{3!}x^3 + \frac{1\cdot4}{6!}x^6 + \frac{1\cdot4\cdot7}{9!}x^9 + \ldots; \tag{2.39}$$

$$g(x) = x + \frac{2}{4!}x^4 + \frac{2\cdot5}{7!}x^7 + \frac{2\cdot5\cdot8}{10!}x^{10} + \ldots \tag{2.40}$$

$$C_1 = 3^{-2/3}/\Gamma(2/3) = 0{,}35502805\ldots;\quad C_2 = 3^{-1/3}/\Gamma(1/3) = 0{,}25881940\ldots$$

МЕТОД РАСЧЕТА

Расчет основан на суммировании приведенных рядов. Последние при $x<0$ являются знакочередующимися и, как отмечалось в комментариях к программе 15, их суммирование на микрокалькуляторе становится невозможным, начиная со сравнительно небольших $|x|$. В данном случае $|x_{\max}| = 7$. Для $x>0$ члены ряда положительны и суммирование возможно, в принципе, до весьма больших x. Одна функция Ai (x) экспоненциально мала при больших x, т. е. член $C_1 f(x)$ и $C_2 g(x)$ при вычислении Ai (x) по формуле (2.37) компенсиру друг друга. Оценки показывают, что существенная потеря точност обусловленная погрешностями округления, наступает при $x >$ Функция Bi (x) при $x > 0$ является, наоборот, *суммой* больших вел чин (см. 2.38), и потери точности не происходит.

Таким образом, изложенный метод применим в интервале $-7 \leqslant x \leqslant 4$ для Ai (x) и при $x \geqslant -7$ для Bi (x).

ПРОГРАММА 21

ФУНКЦИИ ЭЙРИ Ai(x) $(-7 \leqslant x \leqslant 4)$; Bi$(x)$ $(x \geqslant -7)$

	0	1	2	3	4	5
0	В/О 48	F2 22	P4 41	0 04	P5 51	F2 22
1	F/-/ 55	× 26	P3 31	ПП 68	P6 61	F2 22
2	1 14	P2 21	P4 41	/-/ 56	P5 51	F8 82
3	× 26	P6 61	ПП 68	P6 61	F2 22	↑ 06
4	F7 72	× 26	↑ 06	F6 62	− 86	P5 51
5	F6 62	+ 96	3 34	FВП 65	× 26	P6 61
6	С/П 70	F5 52	3 34	+ 96	P5 51	1 14
7	+ 96	× 26	F, 45	↑ 06	F4 42	× 26
8	↑ 06	F3 32	× 26	↑ 06	P4 41	F2 22
9	xy 16	+ 96	P2 21	− 86	PБП 59	P6 61

Регистры памяти				Ai	Bi (x)		
	2	3	4	5	6	7	8
	x					C_1	C_2

Порядок работы

1. *Ввод аргумента* x функций Ai (x) и Bi (x) в <2>.
3. *Ввод* $C_1 = 3{,}5502805 \cdot 10^{-1}$ в <7>.
3. *Ввод* $C_2 = 2{,}588194 \cdot 10^{-1}$ в <8>.
4. *Пуск* В/О С/П .
5. *Результаты*: Ai (x) содержится в регистре <5>, а Bi в регистре <6> и на индикаторе.
6. *Для повторного вычисления* функций при других x доста ввести новое значение x в <2> и затем пуск В/О С/П.

Погрешности:

а) $x < 0$. Функции $\mathrm{Ai}(x)$ и $\mathrm{Bi}(x)$ являются осциллирующими между ± 1. Поэтому далее указаны абсолютные погрешности, которые меньше: $5\cdot 10^{-4}$ при $x \geqslant -7$; $3\cdot 10^{-6}$ при $x \geqslant -6$; $1\cdot 10^{-6}$ при $x \geqslant 4$;

б) $x > 0$. Функция $\mathrm{Ai}(x)$ при больших x монотонно уменьшается, тогда как $\mathrm{Bi}(x)$ монотонно возрастает. В этом случае можно указать относительные погрешности. Для функции $\mathrm{Bi}(x)$ при всех x погрешность меньше $5\cdot 10^{-6}$. Для функции $\mathrm{Ai}(x)$ погрешность меньше: $5\cdot 10^{-3}$ при $x \leqslant 4$; $2\cdot 10^{-4}$ при $x \leqslant 3$; $7\cdot 10^{-6}$ при $x \leqslant 2$.

Время счета возрастает с увеличением $|x|$. Характерные времена видны из приведенных ниже примеров.

Примеры:

Ai (— 7) = 0,1844687. Табличное значение [4] 0,18428084;
Bi (— 7) = 0,2942909. Табличное значение 0,29376207.
Вермя счета 3,5 мин;
Ai (— 5,1) = 0,3095262. Табличное значение 0,30952600;
Bi (— 5,1) = — 0,2120856. Табличное значение — 0,212089.
Время счета 2,5 мин;
Ai (3,831547) = 0,001341. Табличное значение [4] 0,00134139;
Bi (3,831547) = 60,79748. Табличное значение 60,79749.
Время счета 2 мин;
Ai (2,080084) = 0,0308904. Табличное значение 0,03089041;
Bi (2,080084) = 3,648667. Табличное значение 3,648668.
Время счета 1 мин.

2.8. ЭЛЛИПТИЧЕСКИЕ ИНТЕГРАЛЫ

ОПРЕДЕЛЕНИЯ

Эллиптический интеграл первого рода

$$F(\varphi, k) = \int_0^{\varphi} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}}. \qquad (2.41)$$

Эллиптический интеграл второго рода

$$E(\varphi, k) = \int_0^{\varphi} \sqrt{1 - k^2 \sin^2 t}\, dt. \qquad (2.42)$$

Эллиптический интеграл третьего рода

$$\Pi(\varphi, n, k) = \int_0^{\varphi} \frac{dt}{(1 - n \sin^2 t)\sqrt{1 - k^2 \sin^2 t}}. \qquad (2.43)$$

Интегралы, взятые в пределах от 0 до $\pi/2$, называются *полными эллиптическими интегралами*.

Полный эллиптический интеграл первого рода

$$\mathbf{K}\,(k) = F\,(\pi/2,\ k).$$

Полный эллиптический интеграл второго рода

$$\mathbf{E}\,(k) = E\,(\pi/2,\ k).$$

Иногда F, E, **К**, **Е**, П рассматриваются как функции *модулярног* *угла* α, равного по определению

$$\alpha^\circ = \frac{180}{\pi}\arcsin k. \qquad (2.4$$

Обозначим далее

$$\eta = 1 - k^2. \qquad (2.4$$

Согласно (2.44) и (2.45)

$$\eta = \cos^2\,(\pi\alpha/180). \qquad (2.4$$

МЕТОДЫ РАСЧЕТА

Неполные эллиптические интегралы находятся непосредственн вычислением интегралов по формуле Симпсона. Число шагов, требу щееся для обеспечения приемлемой точности ($\sim 10^{-5}$ от полной ве чины), зависит от близости k к 1 и φ к $\pi/2$ (соответствующие оценки да далее).

Полные эллиптические интегралы при указанном методе требу очень большого числа шагов для $k \geqslant 0{,}95$ и соответствующего врем счета. В этом случае значительные преимущества дает применение проксимирующих многочленов. Использование таких многочле гарантирует точность до 10^{-8} при всех k [4]. Однако набор коэффици тов указанных многочленов потребовал бы чрезмерного для микрока кулятора числа программных шагов. Далее используются следую округленные аппроксимирующие многочлены меньшей степени которые дают хорошую точность при $k \geqslant 0{,}9$ и в ряде случаев пр лемую для всех k:

$$\mathbf{K}\,(\eta) = (\ln 4 + 0{,}097933\eta + 0{,}05454\eta^2 + 0{,}032\eta^3) + (0{,}5 + {} + 0{,}124751\eta + 0{,}060112\eta^2 + 0{,}0109\eta^3)\ln\,(1/\eta). \qquad (2$$

$$\mathbf{E}\,(\eta) = (1 + 0{,}444792\eta + 0{,}0851\eta^2 + 0{,}0409\eta^3) + (0{,}2497\eta + {} + 0{,}0815\eta^2 + 0{,}0138\eta^3)\ln\,(1/\eta). \qquad (2$$

Выражение (2.48) дает относительную погрешность не хуже $2\cdot 10^{-5}$ *всех* η, тогда как (2.47) в этом случае гарантирует точность $2\cdot 10^{-4}$

ПРОГРАММА 22

ЭЛЛИПТИЧЕСКИЕ ИНТЕГРАЛЫ ПЕРВОГО И ВТОРОГО РОДА (НЕПОЛНЫЕ И ПОЛНЫЕ) $F(\varphi, k)$, $E(\varphi, k)$, $K(k)$, $E(k)$. ИНТЕГРИРОВАНИЕ ПО ФОРМУЛЕ СИМПСОНА

	0	1	2	3	4	5
0	$\boxed{}$	F7 72	↑ 06	F4 42	÷ 36	P2 21
1	P3 31	− 86	P4 41	1 14	P5 51	4 44
2	ПП 68	F/−/ 55	↑ 06	F4 42	− 86	РПП 69
3	÷ 36	2 24	ПП 68	F/−/ 55	БП 58	Fx⇄y 15
4	ПП 68	5 54	F5 52	↑ 06	F3 32	× 26
5	3 34	÷ 36	P5 51	С/П 78	1 14	P6 61
6	F2 22	P+ 93	↑ 06	F8 82	× 26	F/−/ 55
7	1 14	− 86	/−/ 56	FВП 65	$\boxed{F_,}$ 45	↑ 06
8	F6 62	× 26	↑ 06	F5 52	+ 96	P5 51
9	F2 22	↑ 06	F3 32	+ 96	P2 21	В/О 48

Регистры памяти

			$E(\varphi\ k)$			
2	3	4	5	6	7	8
		N	$F(\varphi, k)$		φ	k

Порядок работы

Программа, помещенная в таблице, *вычисляет* $F(\varphi, k)$ и соответственно $K(k)$. *Для работы с эллиптическими интегралами второго рода, следует заменить команду F, в ячейке 74* на команду Px^y (код 39). Порядок работы для обоих вариантов программ одинаков.

1. *Ввод* φ в <7>. При вычислении полных интегралов в <7> вводится $\pi/2$.
2. *Ввод* k в <8>.
3. *Ввод* N в <4>. Число N равно удвоенному числу шагов интегрирования. Относительная погрешность, меньшая $1 \cdot 10^{-6}$, обеспечивается следующими величинами N (округлять до четного числа):

$$N \approx \begin{cases} 10 + \dfrac{1}{1 - k^2 \sin^2 \varphi} & \text{— для эллиптических интегралов первого рода;} \\ 10 + \dfrac{0{,}4}{1 - k^2 \sin^2 \varphi} & \text{— для эллиптических интегралов второго рода.} \end{cases}$$

Для $k < 0{,}9$ и всех φ достаточно брать $N = 16$. При этом время счета примерно 2 мин.

4. *Пуск* В/О С/П .
5. *Результат*: Выдается на индикатор и хранится в регистре <5>. *Относительная погрешность* меньше $1 \cdot 10^{-6}$. *Время счета* $t \approx N/8$ мин.

Примеры:

$F(\pi/3, \sqrt{3}/2) = 1{,}2125961$. Табличное значение 1,2125966;
$E(\pi/3, \sqrt{3}/2) = 0{,}91839346$. Табличное значение 0,91839329;
$F(\pi/2;\ 0{,}9) = \mathbf{K}(0{,}9) = 2{,}2805492$. Табличное значение 2,28054914;
$E(\pi/2;\ 0{,}9) = \mathbf{E}(0{,}9) = 1{,}1716970$. Табличное значение 1,17169705.
Во всех примерах было взято $N = 16$ и время счета не превышало 2 ми[н].

ПРОГРАММА 23

ПОЛНЫЙ ЭЛЛИПТИЧЕСКИЙ ИНТЕГРАЛ ПЕРВОГО РОДА K(*k*). АППРОКСИМАЦИЯ МНОГОЧЛЕНОМ (2.47)

	0	1	2	3	4	5
0	$\boxed{\text{С/П}\ 78}$	0 04	P, 43	F8 82	8 84	, 46
1	0 04	1 14	6 64	F, 45	× 26	1 14
2	0 04	, 46	2 24	1 14	1 14	F, 45
3	+ 96	P, 43	F8 82	2 24	7 74	7 74
4	F, 45	FВП 65	× 26	3 34	3 34	6 64
5	F, 45	FВП 65	+ 96	P, 43	F8 82	9 94
6	0 04	F, 45	× 26	3 34	1 14	F, 45
7	+ 96	0 04	+ 96	↑ 06	F6 62	× 26
8	↑ 06	P/−/ 53	РБП 59	F7 72	F7 72	FВП 65
9	4 44	× 26	P$\overrightarrow{\overleftarrow{xy}}$ 13	↑ 06	P, 43	+ 96

Регистры памяти	2	3	4	5	6	7	8
					η	$1/\eta$	ln(

Порядок работы

1. *Вычисление* $\eta = 1 - k^2$ и *ввод* в <6>.
2. *Вычисление* $1/\eta$ и $\ln(1/\eta)$ и *ввод* соответственно в <7> и <8>.
3. *Пуск* В/О С/П .
4. *Результат*: Выдается на индикатор K (*k*).

Относительная погрешность меньше: $2 \cdot 10^{-4}$ при всех значе[ниях] k; $3 \cdot 10^{-5}$ при $k \geqslant 0{,}7$; $1 \cdot 10^{-6}$ при $k \geqslant 0{,}9$.

Время счета примерно 20 с.

Примеры:

К (0,1) = 1,574984. Табличное значение [4] 1,5747456;

К (0,9) = 2,280551. Табличное значение 2,28054914.

ПРОГРАММА 24

ПОЛНЫЙ ЭЛЛИПТИЧЕСКИЙ ИНТЕГРАЛ ВТОРОГО РОДА E(k). АППРОКСИМАЦИЯ МНОГОЧЛЕНОМ (2.48)

	0	1	2	3	4	5
0	$\boxed{\text{С/П 78}}$	0 04	P, 43	F8 82	0 04	, 46
1	2 24	4 44	9 94	7 74	× 26	0 04
2	, 46	4 44	4 44	4 44	7 74	9 94
3	2 24	+ 96	P, 43	F8 82	0 04	, 46
4	0 04	8 84	1 14	5 54	× 26	0 04
5	, 46	0 04	8 84	5 54	1 14	+ 96
6	P, 43	F8 82	7 74	2 24	F, 45	× 26
7	5 54	9 94	8 84	F, 45	FВП 65	+ 96
8	0 04	+ 96	↑ 06	F6 62	× 26	↑ 06
9	P/−/ 53	РБП 59	P8 81	P, 43	1 14	+ 96

Регистры памяти	2	3	4	5	6	7	8
					η		$\ln(1/\eta)$

Порядок работы

1. *Ввод* η в <6> ($\eta = 1 - k^2$).
2. *Вычисление* $\ln(1/\eta)$ и *ввод* в <8>.
3. *Пуск* В/О С/П.
4. *Результат:* Выдается на индикатор E (k).

Относительная погрешность меньше: $2 \cdot 10^{-5}$ при всех значениях k; $5 \cdot 10^{-6}$ при $k \geqslant 0{,}5$; $1 \cdot 10^{-6}$ при $k \geqslant 0{,}9$.

Время счета примерно 20 с.

Примеры:

E (0,1) = 1,566852. Табличное значение 1,5668619;

E (0,9) = 1,1716975. Табличное значение 1,17169705.

ПРОГРАММА 25

ЭЛЛИПТИЧЕСКИЙ ИНТЕГРАЛ ТРЕТЬЕГО РОДА П(φ, *n*, *k*). ИНТЕГРИРОВАНИЕ ПО ФОРМУЛЕ СИМПСОНА

	0	1	2	3	4	5
0	$\boxed{\text{В/О} \quad 48}$	F6 62	1 14	P5 51	F4 42	÷ 36
1	/−/ 56	P3 31	− 86	P2 21	ПП 68	F, 45
2	ПП 68	F, 45	+ 96	+ 96	+ 96	P5 51
3	ПП 68	F, 45	+ 96	P5 51	F2 22	РПП 69
4	⟷xy 16	F5 52	3 34	÷ 36	С/П 78	F2 22
5	↑ 06	F3 32	+ 96	P2 21	P+ 93	F/−/ 55
6	↑ 06	P/−/ 53	F8 82	F/−/ 55	/−/ 56	× 26
7	1 14	+ 96	FВП 65	P6 61	P, 43	↑ 06
8	F7 72	× 26	1 14	− 86	↑ 06	F6 62
9	× 26	F, 45	↑ 06	F5 52	+ 96	P5 51

Регистры памяти	2	3	4	5	6	7	8
			1,999995·*N*		φ	*n*	*k*

Порядок работы

1. *Ввод* φ, *n*, *k*, соответственно в <6>, <7>, <8>.

2. *Вычисление* 1,999995·*N* и *ввод* в <4>. Здесь *N* — число шаг при использовании формулы Симпсона. Ориентировочно следует бра

$$\approx 7+0,5/A, \qquad (2.$$

где *A* — наименьшая из величин $(1 - k^2 \sin^2 \varphi)$ и $(1 - n \sin^2 \varphi)$, а следует округлять до целого числа.

Обращаем внимание читателей на то, что число 1,999995 следует н бирать на клавиатуре полностью. Неполный набор числа увеличив погрешность. Округление этого числа ведет к сбоям программы (манды условного перехода).

3. *Пуск* В/О С/П .

4. *Результат*: На индикатор выдается отрицательное число, р ное отношению искомой функции к шагу интегрирования, который слан в регистр <3>. Для получения окончательного результата с дует дополнительно набрать команды ↑ F3 × . После нажа соответствующих клавиш на индикаторе будет искомая величина

Отметим, что в $<5>$ хранится величина $3\Pi/h$, где Π — значение интеграла. Шаг $h < 0$.

Относительная погрешность меньше $1\cdot 10^{-5}$. Для улучшения точности до $1\cdot 10^{-6}$ следует число шагов увеличить вдвое по сравнению с (2.49).

Время счета $t \approx (N/3)$ мин.

Примеры:

$\Pi(\pi/3;\ 0{,}3;\ 0{,}5) = 1{,}2054340$. Табличное значение [4] 1,20543;

$\Pi(5\pi/12;\ 0{,}8;\ \sqrt{3/2}) = 3{,}1684143$. Табличное значение 3,16844;

$\Pi(\pi/2;\ 0{,}9;\ \sin(5\pi/12)) = 12{,}46411$. Табличное значение 12,46407.

В приведенных примерах число шагов согласно (2.49) было взято соответственно 8, 10 и 15. Время счета составило приблизительно 2,5; 3 и 5 мин.

2.9. ОРТОГОНАЛЬНЫЕ МНОГОЧЛЕНЫ (МНОГОЧЛЕНЫ ЧЕБЫШЕВА, МНОГОЧЛЕНЫ И ФУНКЦИИ ЛЕЖАНДРА, МНОГОЧЛЕНЫ ЭРМИТА, ФУНКЦИИ ПАРАБОЛИЧЕСКОГО ЦИЛИНДРА ЦЕЛОГО ПОРЯДКА, МНОГОЧЛЕНЫ ЛАГЕРРА)

Ниже даны программы вычисления значений ортогональных многочленов, перечисленных в заголовке. Каждый из многочленов удовлетворяет соответствующему дифференциальному уравнению и дополнительным условиям, позволяющим выделить данную функцию из класса линейно-независимых решений уравнений. Эти условия могут носить характер разложений в ряды, асимптотических представлений и др. Ниже за основу взяты *определения первых двух многочленов*. Вместе с *рекуррентными формулами*, которые следуют из уравнений, они образуют простой алгоритм для вычисления многочленов.

Многочлены Чебышева $T_n(x)$, $U_n(x)$

Дифференциальное уравнение (совпадает с уравнением для U_n):

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

Рекуррентные формулы:

$$U_n(x) = 2x\,U_{n-1}(x) - U_{n-2}(x); \qquad (2.50)$$
$$T_n(x) = U_n(x) - xU_{n-1}(x).$$

Здесь T_n, U_n — многочлены Чебышева соответственно первого и второго рода. Первый и нулевой многочлены:

$$T_0(x) = 1;\ T_1(x) = x;\ U_0(x) = 1;\ U_1(x) = 2x. \qquad (2.51)$$

Многочлены и функции Лежандра

$P_n(x)$ — многочлены Лежандра (называются также функциями Лежандра первого рода). $Q_n(x)$ — функции Лежандра второго рода.

P_n и Q_n являются линейно-независимыми решениями дифференциального уравнения Лежандра

$$(1-x^2)\frac{d^2 W_n}{dx^2} - 2x\frac{dW_n}{dx} + n(n+1)W_n = 0. \qquad (2.52)$$

Рекуррентная формула

$$W_n(x) = \frac{n-1}{n}\left[\frac{(2n-1)x}{n-1} W_{n-1}(x) - W_{n-2}(x)\right], \qquad (2.53)$$

где W_n означает P_n или Q_n.

Первый и нулевой многочлены и функции:

$$P_0(x) = 1;\ P_1(x) = x;\ Q_0(x) = \frac{1}{2}\ln\frac{1+x}{1-x};\ Q_1(x) = xQ_0(x) - 1. \qquad (2.54)$$

Многочлены Эрмита $H_n(x)$. Функции параболического цилиндра $D_n(x)$

Дифференциальное уравнение для многочленов Эрмита

$$\frac{d^2 H_n}{dx^2} - 2x\frac{dH_n}{dx} + 2nH_n = 0.$$

Рекуррентная формула

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x). \qquad (2.5$$

Первый и второй многочлены:

$$H_0(x) = 1;\ H_1'(x) = 2x. \qquad (2.5$$

Функции параболического цилиндра целого порядка удовлетворяют дифференциальному уравнению

$$d^2D_n/dx^2 + (n + 1/2 - x^2/4)D_n = 0.$$

D_n связана с H_n соотношением

$$D_n(x\sqrt{2}) = 2^{-n/2}e^{-x^2/2}H_n(x). \qquad (2.5$$

Многочлены Лагерра

Дифференциальное уравнение

$$x\frac{d^2 L_n^\alpha(x)}{dx^2} + (\alpha+1-x)\frac{dL_n^\alpha(x)}{dx} + nL_n^\alpha(x) = 0.$$

Рекуррентная формула

$$nL_n^\alpha(x) = (2n+\alpha-1-x)L_{n-1}^\alpha(x) - (n+\alpha-1)L_{n-2}^\alpha(x). \quad (2$$

Нулевой и первый многочлены:

$$L_0^\alpha(x) = 1; \quad L_1^\alpha(x) = \alpha + 1 - x. \qquad (2.59)$$

ПРОГРАММА 26

МНОГОЧЛЕНЫ ЧЕБЫШЕВА ПЕРВОГО И ВТОРОГО РОДА $T_n(x)$, $U_n(x)$. РЕКУРРЕНТНЫЕ ФОРМУЛЫ (2.50) И СООТНОШЕНИЯ (2.51)

	0	1	2	3	4	5
0	$\boxed{}$	2 24	P4 41	1 14	P6 61	F2 22
1	2 24	× 26	P5 51	F5 52	↑ 06	P, 43
2	F2 22	× 26	2 24	× 26	↑ 06	F6 62
3	− 86	P5 51	P/−/ 53	P6 61	F4 42	1 14
4	+ 96	P4 41	F3 32	− 86	РБП 59	P xy 13
5	F6 62	↑ 06	F2 22	× 26	/−/ 56	↑ 06
6	F5 52	+ 96	P6 61	С/П 78		
7						
8						
9						

Регистры памяти	2	3	4	5	6	7	8
	x	n		$U_n(x)$	$T_n(x)$		

Порядок работы

1. *Ввод аргумента* x многочленов T_n и U_n в <2>.
2. *Ввод порядка* n в <3>, $n \geqslant 2$; многочлены с $n = 0$; 1 определены соотношениями (2.51).
3. *Пуск* В/О С/П .
4. *Результаты*: $T_n(x)$ в регистре <6>; $U_n(x)$ в регистре <5>.

Погрешности. Если модуль вычисленного значения функции меньше 0,1, то *абсолютная погрешность не превышает* $5 \cdot 10^{-7}$. В остальных случаях *относительная погрешность* меньше $5 \cdot 10^{-6}$.

Время счета $t \approx (n/15)$ мин, где n — порядок многочлена.

Примеры:

$T_{12}(0{,}6) = 0{,}1315858$. Табличное значение [4] 0,13158560;
$U_{12}(0{,}6) = -0{,}6118927$. Табличное значение $-0{,}6118929$.

ПРОГРАММА 27

МНОГОЧЛЕНЫ ЛЕЖАНДРА $P_n(x)$, ФУНКЦИИ ЛЕЖАНДРА ВТОРОГО РОДА $Q_n(x)$. РЕКУРРЕНТНЫЕ ФОРМУЛЫ (2.53) И СООТНОШЕНИЯ (2.54)

	0	1	2	3	4	5
0	$\boxed{\text{В/О}\ 48}$	1 14	Р6 61	F2 22	ПП 68	Р, 43
1	F5 52	Р8 81	F2 22	1 14	+ 96	2 24
2	− 86	÷ 36	/−/ 56	Р $\overleftarrow{xy}\rightarrow$ 13	2 24	÷ 36
3	Р6 61	↑ 06	F2 22	× 26	1 14	− 86
4	ПП 68	Р, 43	С/П 78	Р5 51	2 24	Р4 41
5	F4 42	1 14	− 86	÷ 36	Р7 71	1 14
6	+ 96	↑ 06	F2 22	× 26	↑ 06	F5 52
7	× 26	↑ 06	F6 62	− 86	↑ 06	F5 52
8	Р6 61	F7 72	÷ 36	Р5 51	F4 42	1 14
9	÷ 96	Р4 41	F3 32	− 86	РБП 59	, 46

				$Q_n(x)$			$P_n(x)$
Регистры памяти	2	3	4	5	6	7	8
	x	n					

Порядок работы.

1. *Ввод аргумента* x в <2>.
2. *Ввод порядка* n в <3>, $n \geqslant 2$; многочлены и функции с $n = 0$; определены соотношениями (2.54).
3. *Пуск* В/О С/П .
4. *Результаты*: $P_n(x)$—в регистре <8>; $Q_n(x)$—в регистре <5

Погрешности. При $x < 1$ применение рекуррентных формул д возрастающих значений n дает небольшую потерю точности [4]. Д обеих функций *относительная погрешность* меньше $5 \cdot 10^{-6}$, когда в численное значение функции по модулю меньше 0,1. В противном сл чае следует пользоваться абсолютной погрешностью, которая мень $5 \cdot 10^{-7}$. При $x > 1$ выражения (2.54) для Q_0 и Q_1 теряют смысл, но ко ректность алгоритма вычисления $P_n(x)$ и оценки погрешности п $x < 1$ сохраняются. При задании $x > 1$ происходит автоматическ останов за счет операции ln $(1 - x)$, но перед этим в регистр <8> *зас лается правильное значение* $P_n(x)$.

Время счета $t \approx (n/6)$ мин.

Примеры:

$P_8(0{,}49) = -0{,}4408775$. Табличное значение [4] $-0{,}4408775$;
$Q_8(0{,}49) = -0{,}1699204$. Табличное значение $-0{,}16992027$;
$P_9(0{,}97) = 0{,}03750364$. Табличное значение $0{,}03750397$;
$Q_9(0{,}97) = -0{,}81464728$. Табличное значение $-0{,}81464729$;
$P_{10}(10) = 1{,}761875 \cdot 10^{12}$. Табличное значение $1{,}76188 \cdot 10^{13}$.

ПРОГРАММА 28

МНОГОЧЛЕНЫ ЭРМИТА $H_n(x)$. ФУНКЦИИ ПАРАБОЛИЧЕСКОГО ЦИЛИНДРА ЦЕЛОГО ПОРЯДКА $D_n(x)$. РЕКУРРЕНТНЫЕ ФОРМУЛЫ (2.55) И СООТНОШЕНИЯ (2.56), (2.57)

	0	1	2	3	4	5
0	$\boxed{}$	2 24	P4 41	1 14	P6 61	F2 22
1	2 24	× 26	P5 51	F4 42	1 14	− 86
2	↑ 06	F6 62	× 26	P8 81	F2 22	↑ 06
3	F5 52	P6 61	× 26	↑ 06	F8 82	− 86
4	2 24	× 26	P5 51	F4 42	1 14	+ 96
5	P4 41	F3 32	− 86	РБП 59	$P\overleftarrow{\underrightarrow{xy}}$ 13	F3 32
6	2 24	$P\overleftarrow{\underrightarrow{xy}}$ 13	× 26	↑ 06	F2 22	F/−/ 55
7	+ 96	2 24	÷ 36	/−/ 56	P÷ 33	↑ 06
8	F5 52	× 26	P6 61	С/П 78		
9						

D_n	$x/\sqrt{2}$	n		$D_n(x)$			
Регистры памяти	2	3	4	5	6	7	8
H_n	x	n		$H_n(x)$			

Порядок работы

1. *Ввод* n в <3>, $n \geqslant 2$. Для $n = 0,1$ (см. (2.56), (2.57)).
2. а) *Ввод* x в <2> при расчете $H_n(x)$.
   б) *Ввод* $x/\sqrt{2}$ в <2> при расчете $D_n(x)$.
3. *Пуск* В/О С/П.
4. *Результаты*:
   а) $H_n(x)$ в <5> при вводе в <2> x.
   б) $D_n(x)$ в <6> при вводе в <2> $x/\sqrt{2}$.

Погрешности. Для $H_n(x)$ и $D_n(x)$ *относительные погрешности* меньше соответственно $1 \cdot 10^{-6}$ и $5 \cdot 10^{-6}$, когда модуль вычисленного значения функции, стоящей в регистре <5>, больше 0,1. В противном случае *абсолютная погрешность* $H_n(x)$ меньше $1 \cdot 10^{-7}$. Относительную погрешность функции $D_n(x)$ в этом случае можно оценить как относительное приращение числа в регистре <5> при изменении его величины на $1 \cdot 10^{-6}$.

Время счета $t \approx (n/10)$ мин.

Примеры:

$H_{11}(10) = 1{,}533736 \cdot 10^{14}$. Табличное значение [4] $1{,}5337306 \cdot 10^{14}$;

$D_4(5) = 0{,}9227568$. Табличное значение 0,92276 (дано в [4] до 5 значащих цифр).

ПРОГРАММА 29
МНОГОЧЛЕНЫ ЛАГЕРРА $L_n^\alpha(x)$

	0	1	2	3	4	5
0		2 24	P7 71	F4 42	1 14	P6 61
1	+ 96	↑ 06	F2 22	− 86	P5 51	F4 42
2	↑ 06	F7 72	+ 96	1 14	− 86	P8 81
3	F7 72	↑ 06	F2 22	− 86	↑ 06	F8 82
4	÷ 36	1 14	+ 96	↑ 06	F5 52	× 26
5	↑ 06	F6 62	− 86	↑ 06	F5 52	P6 61
6	F8 82	× 26	↑ 06	F7 72	÷ 36	P5 51
7	F7 72	1 14	+ 96	P7 71	F3 32	− 86
8	РБП 59	$F\overleftarrow{xy}$ 15	С/П 78			
9						

Регистры памяти	2	3	4	5	6	7	8
	x	n	α	$L_n^\alpha(x)$			

Порядок работы

1. *Ввод аргумента* x в <2>.
2. *Ввод порядка* n в <3>, $n \geqslant 2$. Для $n = 0$; 1 (см. (2.59)).
3. *Ввод параметра* α в <4>.
4. *Пуск* В/О С/П
5. *Результат*: $L_n^\alpha(x)$ в регистре <5>.

Погрешности. Относительная погрешность меньше $2 \cdot 10^{-6}$, когда вычисленное значение функций по модулю больше 0,1. В противном случае следует пользоваться *абсолютной погрешностью*, которая меньше $2 \cdot 10^{-7}$.

Время счета $t \cong (n/10)$ мин.

Примеры:

$L_{10}^{15}(0) = 3{,}26875 \cdot 10^6$. Табличное значение $L_{10}^{15}(0) = \dfrac{25!}{15! \cdot 10!}$, что составляет $3{,}268760 \cdot 10^6$;

$L_9^{-9}(12) = -1{,}42189 \cdot 10^4$. Точное значение $L_9^{-9}(12) = -\dfrac{12^9}{9!} = -1{,}421897 \times 10^4$.

Глава 3

Программы общего назначения

3.1. ВЫЧИСЛЕНИЕ ОПРЕДЕЛЕННЫХ ИНТЕГРАЛОВ ПО ФОРМУЛАМ СИМПСОНА И ТРАПЕЦИЙ

Формула Симпсона (см., например, [9]) имеет вид

$$F = \int_{x_0}^{L} f(x)\,dx = \frac{h}{3}\left[f(x_0) + \sum_{k=0}^{N-1}\{4f[x_0 + (2k+1)h] + 2\varepsilon f(x_0 + 2kh)\} + f(L)\right], \tag{3.1}$$

где шаг интегрирования $h = (L - x_0)/(2N)$; $\varepsilon = 0$; 1 соответственно для $k = 0$ и $k > 0$.

Формула трапеций

$$F = \int_{x_0}^{L} f(x)\,dx = \frac{h}{2}\left[f(x_0) + \sum_{k=1}^{N-1} 2f(x_0 + kh) + f(L)\right]. \tag{3.2}$$

где шаг интегрирования $h = (L - x_0)/N$.

Формула (3.1) дает четвертый порядок точности относительно h, тогда как (3.2) — только второй. Даже если учесть, что подынтегральная функция $f(x)$ при использовании (3.1) должна вычисляться $(2N + 1)$ раз на интервале $x_0 \div L$, а при использовании (3.2) только N раз, время счета по формуле Симпсона при достижении одинаковой точности оказывается часто существенно меньшим, чем по формуле трапеций.

При выборе того или иного варианта программы численного интегрирования для реализации на микрокалькуляторе нужно, однако, принимать во внимание и следующее:

а) число программных шагов и свободных регистров памяти, которые можно использовать в подпрограмме $f(x)$;

б) удобство работы с программой (простота ввода исходных данных и вывода результатов).

Наибольшее значение имеет п. а), с которым связана возможность вообще постановки соответствующих задач на микрокалькуляторе при достаточно сложном виде подынтегральной функции. В таких слу-

5 Зак. 1219

чаях применение формулы трапеций, которая проще формулы Симпсо-на, может оказаться выходом из положения.

Число шагов, которые могут быть отведены на программирование $f(x)$ (далее эта величина для краткости называется «длиной $f(x)$») можно увеличить, сократив программу путем исключения из нее рас-чета $f(x_0)$ или $f(L)$ или даже обеих этих функций. Последние входят в качестве слагаемых в (3.1) и (3.2). Они могут быть вычислены отдель-но и введены в качестве исходных данных. Подобный прием целесооб-разен, когда граничные значения $f(x)$ вычисляются достаточно просто. В некоторых случаях на одном конце или на обоих концах интегриро-вания функция $f(x)$ имеет устранимые особенности, и соответствующие значения должны быть найдены предельным переходом. Тогда отдель-ный ввод $f(x_0)$ и $f(L)$ практически необходим.

Увеличение длины $f(x)$ при отдельном вводе граничных значений оказывается довольно большим. Так, программа 32, в которой вводит-ся сумма $f(x_0) + f(L)$, характеризуется длиной $f(x)$ в 37 шагов, тогда как для аналогичной программы 29, куда включен расчет $f(x_0)$ и $f(L)$, длина $f(x)$ составляет 27 шагов. В программе 33, использующей фор-мулу трапеций и отдельный ввод $f(x_0) + f(L)$, длина $f(x)$ составляет 45 шагов, т. е. 3/4 полной программной памяти микрокалькулятора.

Формулы Симпсона и трапеций содержат операции умножения на $h/3$ и $h/2$ соответственно (см. (3.1) и (3.2)). Последние, хотя и выполня-ются один раз, требуют 6 программных шагов и соответственно сокра-щают длину $f(x)$. Поэтому их целесообразно выполнить отдельно (вне программы) как заключительные после завершения автоматическог[о] счета по программе. Отметим, что если длина $f(x)$ мала и остаются лиш-ние 6 шагов и, кроме того, если счет прекращается по команде С/П, легко скорректировать программу, введя перед командой С/П указан-ные операции, изменив адресацию обращения к подпрограмме $f(x)$.

Далее приведено несколько вариантов программ вычисления опре-деленных интегралов. Каждая помещена в универсальную таблицу (см. гл. 1), где обозначены пустые ячейки для записи подпрограммы подын-тегральной функции. Здесь же указана максимальная длина $f(x)$.

Целесообразно сначала составить подпрограмму $f(x)$ и после этого выбрать подходящий вариант программы. В комментариях к каждо[й] программе, которые приведены после соответствующей таблицы, пред-полагается, что *основная программа уже набрана и перед запуском* ав-томатического счета остается *набрать подпрограмму вычисления $f(x)$ и ввести исходные данные.*

Ввод подпрограммы начинается с первой ячейки отведенного дл[я] нее массива программной памяти (обведенный пустой участок в таблице). Направить команду в указанную ячейку в режиме программирования можно или несколькими нажатиями клавиш $\overleftarrow{\text{ШГ}}$ или $\overrightarrow{\text{ШГ}}$, если теку-щий адрес близок к адресу нужной ячейки, или переводом калькулято-ра в рабочий режим ($\overrightarrow{\text{РШГ}}$). После этого нужно набрать команду безу-словного перехода БП А и возвратить калькулятор в режим програм-мирования ($\overleftarrow{\text{РШГ}}$). Первая команда, набранная после этого, окажется в нужной ячейке. Здесь А — адресная команда первой ячейки подпрограммы. Нужные адресные команды указаны в комментариях к программам следующим образом: «... *начало подпрограммы* БП А».

Если длина выделенной части подпрограммы короче максимальной более чем на один шаг (*в комментариях такие подпрограммы названы «короткими»*), то в ее конце целесообразно добавить команду условного перехода на стандартное окончание подпрограммы. Это добавление необязательно, оно несколько сокращает время счета и способствует устранению возможных сбоев, так как при этом обходятся пустые ячейки подпрограммы, куда могли быть случайно занесены посторонние команды. Нужные команды безусловного перехода также указаны в комментариях.

В отличие от предыдущих программ значения погрешностей едва ли могут быть указаны, даже если задан конкретный вид $f(x)$. Также трудно заранее дать рекомендации по выбору числа шагов интегрирования N. Распространенным способом является удвоение N до тех пор, пока разность между вычисленными значениями интеграла на двух соседних итерациях не станет близкой к заданному значению погрешности. Поскольку формула Симпсона имеет четвертый порядок точности, то можно ожидать, что при каждом удвоении N погрешность уменьшается примерно в 16 раз. Для формулы трапеции характерное уменьшение погрешности составляет половину порядка при удвоении N.

Отметим, что когда подынтегральная функция имеет вид многочлена степени n ($n < 10$) и используется формула Симпсона, число шагов $N < 25$. При этом погрешность не превышает единицы в шестой значащей цифре.

ПРОГРАММА 30

ИНТЕГРАЛ $F = \int_{x_0}^{b} f(x)dx$. ФОРМУЛА СИМПСОНА (3.1)

	0	1	2	3	4	5
0	$\boxed{\text{В/О 48}}$	ПП 68	÷ 36	ПП 68	÷ 36	+ 96
1	+ 96	+ 96	Р5 51	F2 22	↑ 06	F4 42
2	− 86	РПП 69	Р÷ 33	ПП 68	÷ 36	+ 96
3	Р5 51	БП 58	Р↑ 03	ПП 68	÷ 36	С/П 78
4	F2 22	↑ 06	F3 32	+ 96	Р2 21	
5						
6			f(x)	27 шагов		
7						
8						
9			↑ 06	F5 52	+ 96	Р5 51

				$3F/h$	$f(x)$		
Регистры памяти	2	3	4	5	6	7	8
	x_0-h	h	$L-2h$	0			

Порядок работы

1. *Программирование* $f(x)$. Начало подпрограммы БП F, . Дл программирования $f(x)$ свободны регистры памяти <6>, <7>, <8> Текущее значение аргумента x подынтегральной функции хранитс в регистре <2>. Кроме того, перед обращением к подпрограмме вели чина x попадает в регистр <X>. В этом же регистре должен остав ляться к концу подпрограммы ее результат, т. е. вычисленное значе ние $f(x)$. При короткой программе (см. вводную часть данного пар графа) в ее конце следует поставить команду условного переход БП F9 .

2. *Вычисление шага* $h = (L - x_0)/(2N)$ *и ввод* в <3>.

3. *Вычисление* $(x_0 - h)$ *и ввод* в <2>.

4. *Вычисление* $(L - 2h)$ *и ввод* в <4>.

5. *Ввод* 0 в <5>.

6. *Пуск* В/О С/П .

7. *Результат*: В регистре <5> величина $3F/h$. Искомое значени F получается умножением содержимого <5> на $h/3$. Операция може быть выполнена с помощью следующей системы команд $F5$ ↑ F × 3 ÷ . После этих команд на индикаторе оказывается вел чина F.

Пример. Вычислить $\int_0^2 10x^9\,dx$.

Длина подпрограммы подынтегральной функции 9 шагов: F/—/ F/— F/—/ ↑ $F2$ × 1 0 × . Отметим, что первой операцией является воз дение x в квадрат, так как величина x, как указывалось в п. 1, перед обращени к подпрограмме уже находится в <X>. (В приведенной подпрограмме x^9 вы сляется как $((x^2)^2)^2 \cdot x$. Подобная замена операции возведения в целую степе умножением и возведением в квадрат дает выигрыш в точности. Встроен функция x^y вычисляет x^n как $e^{n \ln x}$. Она дает бо́льшие погрешности, чем дру встроенные функции и неприменима при отрицательных x). Добавим к подп грамме команду безусловного перехода БП $F9$ (см. § 3.1).

Результаты:

$F = 1112{,}968$, время счета 0,5 мин при $N = 2$;

$F = 1030{,}617$, время счета 1 мин при $N = 4$;

$F = 1024{,}431$, время счета 1,5 мин при $N = 8$;

$F = 1024{,}027$ время счета 3 мин при $N = 16$;

$F = 1024{,}0015$, время счета 6 мин при $N = 32$.

Точное значение $F = 2^{10} = 1024$.

ПРОГРАММА 31

ИНТЕГРАЛ $F=\int\limits_{x_0}^{L} f(x)dx$. ФОРМУЛА СИМПСОНА (3.1). ОТДЕЛЬНЫЙ ВВОД $f(x_0)$; x_0 МОЖЕТ БЫТЬ ТОЧКОЙ УСТРАНИМОГО РАЗРЫВА $f(x)$

	0	1	2	3	4	5
0	$\boxed{\text{В/О}\ 48}$	ПП 68	F3 32	ПП 68	F3 32	+ 96
1	+ 96	+ 96	P5 51	F2 22	↑ 06	F4 42
2	− 86	FВП 65	ПП 68	F3 32	+ 96	P5 51
3	БП 58	P↑ 03	F2 22	↑ 06	F3 32	− 86
4	P2 21					
5						
6			f(x)	31 шаг		
7						
8						
9			↑ 06	F5 52	+ 96	P5 51

				$3F/h$	$f(x)$		
Регистры памяти	2	3	4	5	6	7	8
	$L+h$	h	x_0+2h	$f(x_0)$			

Порядок работы

1. *Программирование* $f(x)$. Начало подпрограммы БПР4. Для программирования $f(x)$ свободны регистры <6>, <7>, <8>. Текущее значение аргумента x хранится в регистре <2>. Кроме того, в основной программе перед обращением к подпрограмме x помещается в регистр <X>. В этом же регистре должен оставляться к концу подпрограммы ее результат, т. е. вычисленное значение $f(x)$. При короткой подпрограмме добавляется в ее конце команда условного перехода БП F9.
2. *Вычисление шага* $h=(L-x_0)/(2N)$ и *ввод* в <3>.
3. *Вычисление* $(L+h)$ и *ввод* в <2>.
4. *Вычисление* (x_0+2h) и *ввод* в <4>.
5. *Вычисление* $f(x_0)$ или $\lim\limits_{x\to x_0}[f(x)]=f(x_0)$ и *ввод* в <5>.
6. *Пуск* В/О С/П. При останове — индикация переполнения.

7. *Результат*: В регистре <5> — величина 3 F/h. Искомое значение F получается умножением содержимого <5> на $h/3$ (команды $F5$ ↑ $F3$ × 3 ÷). После выполнения указанных команд на индикаторе — величина F.

Пример. Вычислить $S(1) = \int_0^1 \frac{\sin x}{x}\,dx$ (интегральный синус).

Длина подпрограммы $\sin x/x$ 4 шага; Р+ ↑ $F2$ ÷ . Величина $f(x_0) = \lim\limits_{x \to 0} (\sin x/x) = 1$.

Результаты:

$S(1) = 0{,}94608690$, время счета примерно 0,5 мин при $N = 2$;

$S(1) = 0{,}94608336$, время счета примерно 1 мин при $N = 4$;

$S(1) = 0{,}94608333$, время счета примерно 1,5 мин при $N = 8$.

Как видно, увеличение N с 4 до 8 практически ничего не меняет. Таким образом, можно принять, что $S(1) = 0{,}9460833$. Табличное значение $S(1) = 0{,}94608307$.

ПРОГРАММА 32

ИНТЕГРАЛ $F = \int_0^L f(x)dx$. ФОРМУЛА СИМПСОНА (3.1). ОТДЕЛЬНЫЙ ВВОД $f(0)$. ПРЕДЕЛЫ ИНТЕГРАЛА $x=0$, L ЯВЛЯЮТСЯ РЕГУЛЯРНЫМИ ТОЧКАМИ $f(x)$

Длина $f(x)$ при указанных условиях возрастает до 33 шагов. Кроме того, к подпрограмме $f(x)$ добавляется 1 регистр памяти, и, наконец, останов производится по команде С/П, а не по переполнению (как в предыдущей программе). Это делает программу более гибкой (см. с. 66).

	0	1	2	3	4	5
0	$\boxed{\text{В/0 48}}$	ПП 68	Р× 23	ПП 68	Р× 23	+ 96
1	+ 96	+ 96	Р4 41	ПП 68	Р× 23	+ 96
2	Р4 41	БП 58	Р↑ 03	F2 22	↑ 06	F3 32
3	− 86	Р2 21	РПП 69	F÷ 35	С/П 78	
4						
5						
6			f(x)	33 шага		
7						
8						
9			↑ 06	F4 42	+ 96	Р4 41

			$3F/h$		$f(x)$		
Регистры памяти	2	3	4	5	6	7	8
	$L+h$	h	$-f(0)$				

Порядок работы

1. *Программирование* $f(x)$. Начало подпрограммы БП $F\div$. Свободны регистры <5>, <6>, <7>, <8>. Переменную x следует брать из <2>. Кроме того, перед обращением к подпрограмме x помещается в регистр <X>. В этом же регистре оставляется результирующая величина $f(x)$. При короткой программе добавляется команда БП $F9$.

2. *Вычисление шага* $h = L/(2,000001 \cdot N)$ и *ввод* в <3>.

Округление знаменателя недопустимо, так как может вызвать сбой программы.

3. *Вычисление* $(L+h)$ и *ввод* в <2>.

4. *Вычисление* $-f(0)$ и *ввод* в <4>.

5. *Пуск* В/О С/П.

6. *Результат*: В регистре <4> величина $3F/h$. Домножить ее на $h/3$ возможно с помощью следующей системы команд:
$F4$ ↑ $F3$ × 3 ÷ . На индикатор выдается величина F.

Пример. Вычислить $F = \int\limits_0^{\infty} e^{-2\,\mathrm{ch}\,x}dx$ — модифицированную функцию Бесселя $K_0(2)$ [4].

Примем в качестве верхнего предела интеграла $L = 3$. При $L = 3$ подынтегральная функция меньше $1 \cdot 10^{-8}$, что обеспечивает достаточное приближение к несобственному интегралу. Длина подпрограммы $f(x) = e^{-2\,\mathrm{ch}\,x}$ составляет 6 шагов: Р ÷ ↑ F, + /—/ Р ÷ . Вычисляем и вводим в регистр <4> в качестве исходной величины $-f(0) = -e^{-2}$.

Результаты:

$K_0(2) = 0{,}1134951$, время счета 40 с при $N = 2$;

$K_0(2) = 0{,}11389461$, время счета 1,5 мин при $N = 4$;

$K_0(2) = 0{,}11389375$, время счета 2,5 мин при $N = 8$.

Судя по приведенным данным, дальнейшее увеличение N не приведет к существенному изменению последнего результата. Таким образом, можно принять $K_0(2) = 0{,}1138938$. Табличное значение $K_0(2) = 0{,}11389387$.

ПРОГРАММА 33

ИНТЕГРАЛ $F = \int_{x_0}^{L} f(x)dx$. ФОРМУЛА СИМПСОНА (3.1). ОТДЕЛЬНЫЙ ВВОД $[f(x_0)+f(L)]/2$; x_0, L МОГУТ БЫТЬ ТОЧКАМИ УСТРАНИМОГО РАЗРЫВА $f(x)$

	0	1	2	3	4	5
0	В/О 40	ПП 68	F2 22	+ 96	Р5 51	F4 42
1	↑ 06	F2 22	− 86	FВП 65	ПП 68	F2 22
2	БП 58	Р0 01	F2 22	↑ 06	F3 32	+ 96
3	Р2 21					
4						
5			$f(x)$	37 шагов		
6						
7						
8						
9			↑ 06	F5 52	+ 96	Р5 51

			$3F/2h$		$f(x)$	
Регистры памяти 2	3	4	5	6	7	8
x_0	h	$L-2h$	$\frac{f(x_0)+f(L)}{2}$			

Порядок работы

1. *Программирование* $f(x)$. Начало подпрограммы БП Р3 . Дл подпрограммы $f(x)$ свободны регистры <6>, <7>, <8>. Перемен ную x следует брать из <2>. Кроме того, к началу подпрограммы введена в регистр <X>. В этом же регистре оставляется результирую щая величина $f(x)$, которая подготавливается подпрограммой. Пр короткой подпрограмме $f(x)$ добавляется команда БП F9 .
2. *Вычисление шага* $h = (L - x_0)/(2N)$ и *ввод* в <3>.
3. *Ввод* x_0 в <2>.
4. *Вычисление* $(L - 2h)$ и ввод в <4>.
5. *Вычисление полусуммы краевых значений* $[f(x_0) + f(L)]/2$ и *вво* в <5>.
6. *Пуск* В/О С/П . Останов — по переполнению.

7. *Результат*: В регистре <5> величина $3F/(2h)$. Она домножается на $2h/3$ с помощью следующей системы команд $F5$ ↑ $F3$ × 2 × 3 ÷ . Величина F выдается на индикатор.

Пример. Вычислить $F = \frac{1}{\pi}\int_0^\pi \cos(3 \sin x)\,dx$ — функцию Бесселя первого рода $J_0(3)$.

Длина подпрограммы $\frac{1}{\pi}\cos(3 \sin x)$ составляет 7 шагов: $P+$ 3 × P— ↑ P× ÷ . В качестве исходной величины вводим в <5> $[f(0)+f(\pi)]/2 = 1/\pi$.

Результаты:

$J_0(3) = -0{,}34708796$, время счета 40 с при $N = 2$;

$J_0(3) = -0{,}26038092$, время счета 1 мин при $N = 4$;

$J_0(3) = -0{,}26005226$, время счета 2 мин при $N = 8$;

$J_0(3) = -0{,}26005164$, время счета 4,5 мин при $N = 16$.

Судя по приведенным данным, дальнейшего увеличения N не требуется. Можно принять $J_0(3) = -0{,}260052$. Табличное значение $J_0(3) = -0{,}26005195...$ [4].

ПРОГРАММА 34

ИНТЕГРАЛ $F = \int_{x_0}^{L} f(x)dx$. ФОРМУЛА ТРАПЕЦИЙ (3.2). ОТДЕЛЬНЫЙ ВВОД $f(x_0)+f(L)$; x_0 и L МОГУТ БЫТЬ ТОЧКАМИ УСТРАНИМОГО РАЗРЫВА $f(x)$

	0	1	2	3	4	5
0	С/П 78					
1						
2						
3			$f(x)$	45 шагов		
4						
5						
6						
7					↑ 06	F5 52
8	+ 96	P5 51	F2 22	↑ 06	F3 32	+ 96
9	P2 21	↑ 06	F4 42	— 86	PB/0 49	P0 01

Регистры памяти	2	3	4	5	6	7	8
				F/h	$f(x)$		
	x_0+h	h	$L-0{,}1h$	$\frac{f(x_0)+f(L)}{2}$			

Порядок работы

1. *Программирование* $f(x)$. Начало подпрограммы В/О (перва ячейка). Для подпрограммы $f(x)$ свободны регистры $<6>$, $<7>$, $<8>$. Переменную x следует брать из $<2>$. Результирующая вели чина $f(x)$, которая подготавливается подпрограммой, оставляется регистре $<X>$. При короткой подпрограмме $f(x)$ добавляется коман да БП 7 .

2. *Вычисление шага* $h = (L - x_0)/N$ и *ввод* в $<3>$.

3. *Вычисление* $(x_0 + h)$ и *ввод* в $<2>$.

4. *Вычисление* $(L - 0{,}1h)$ и *ввод* в $<4>$.

5. *Вычисление* $[f(x_0) + f(L)]/2$ и *ввод* в $<5>$.

6. *Пуск* В/О С/П .

7. *Результат*: В регистре $<5>$ величина F/h. Она домножаетс на h, что осуществляется с помощью следующей системы команд $F5$ $\uparrow$ $F3$ $\times$. Величина F выдается на индикатор.

Примеры: 1. Вычислить $J_0(3) = \frac{1}{\pi}\int\limits_0^\pi \cos(3 \sin x)\, dx$ — пример к предыд щей программе.

Подпрограмма $\frac{1}{\pi}\cos(3 \sin x)$: $F2$ Р+ 3 × Р— ↑ Р× ÷ . П сравнению с примером к программе 33 здесь добавлена первая команда $F2$. В о личие от программы 32 (как и других программ, использующих формулу Симпс на) переменная x не содержится в регистре $<X>$ перед началом подпрогра мы $f(x)$ и ее необходимо ввести в $<X>$ из $<2>$.

Результатыты:

$J_0(3) = -0{,}25906506$, время счета 0,5 мин при $N = 4$;
$J_0(3) = -0{,}26005195$, время счета 1 мин при $N = 8$;
$J_0(3) = -0{,}26005226$, время счета 2 мин при $N = 16$.

Сравнение с примером к программе 33 показывает, что для достижения о наковой точности формула трапеций требует даже меньшего времени, чем фор ла Симпсона. Причина этого, очевидно, связана со сравнительно плавным изм нением подынтегральной функции.

2. Вычислить $F = \int\limits_0^2 10x^9 dx$ (пример к программе 30).

Результаты вычислений:

$F = 1477{,}2265$, время счета 0,5 мин при $N = 4$;
$F = 1053{,}8909$, время счета 2 мин при $N = 16$;
$F = 1025{,}8744$, время счета 8 мин при $N = 64$;
$F = 1024{,}4685$, время счета 17 мин при $N = 128$.

Напомним, что точное значение интеграла $F = 2^{10} = 1024$. Видно, что стижение высокой точности применением формулы трапеций при столь быстр изменении функции как x^9 связано с чрезмерными затратами времени. Здесь фективность формулы Симпсона существенно выше (см. пример к программе 3

3.2. ВЫЧИСЛЕНИЕ НЕКОТОРЫХ НЕСОБСТВЕННЫХ ИНТЕГРАЛОВ

Далее приведены программы вычисления несобственных интег лов вида

$$F = \int\limits_a^b \frac{\varphi(x)\, dx}{(x-a)^m (b-x)^n}, \qquad (3$$

где $m \leqslant 1/2$; $n \leqslant 1/2$. Функция $\varphi(x)$ — непрерывная на (a, b) за ключением устранимых разрывов в $x = a, b$ и в отдельных внутрен точках, которые не должны совпадать с узлами интегрирования $x = x_0 + kh$ (см. (3.1), (3.2)).

Без ограничения общности можно положить $a = 0$ и перейти от (3.3) к интегралу

$$F = \int_0^L \frac{\varphi(x)\,dx}{x^m (L-x)^n}. \tag{3.4}$$

Подстановка $x = L\cos^2\vartheta$ приводит (3.4) к виду

$$F(L) = 2L^{1-m-n}\int_0^{\pi/2} \varphi(L\cos^2\vartheta)(\cos\vartheta)^{1-2m}(\sin\vartheta)^{1-2n}\,d\vartheta. \tag{3.5}$$

Если $\varphi(x)$ регулярна, то в отличие от (3.4) подынтегральная функция в (3.5) не имеет особенностей. Однако непосредственное применение формул Симпсона или трапеций к (3.5) при умеренном числе шагов, в общем, дает значения $F(L)$ с неудовлетворительной точностью. Особенно это относится к формуле трапеций (см. далее примеры к программам). Причиной является быстрое изменение подынтегрального выражения в (3.5) из-за сомножителей $(\cos\vartheta)^{1-2m}$ и $(\sin\vartheta)^{1-2n}$. Указанные функции имеют бесконечные производные на концах интервала интегрирования.

Для улучшения точности суммы типа (3.1) или (3.2) можно дополнить интегралами от линеаризованных быстро меняющихся подынтегральных функций на первом и последнем шаге в формуле трапеций или на первом и последнем удвоенном шаге в формуле Симпсона. Медленно меняющаяся часть подынтегрального выражения и, в частности, $\varphi(L\cos^2\vartheta)$ полагается равной граничному значению, т. е. соответственно $\varphi(L)$ или $\varphi(0)$. В результате скорректированная формула Симпсона принимает вид

$$F(L) = 2L^{1-m-n}\Bigg\{\varphi(L)\int_0^{2h}\vartheta^{1-2n}\,d\vartheta - \frac{h}{3}\varphi(L)\left[4h^{1-2n} + (2h)^{1-2n}\right] + \frac{h}{3}\Sigma^{(c)} + \varphi(0)\int_0^{2h}\vartheta^{1-2m}\,d\vartheta - \frac{h}{3}\varphi(0)\left[4h^{1-2m} + (2h)^{1-2m}\right]\Bigg\}, \tag{3.6}$$

где $\frac{h}{3}\Sigma^{(c)}$ означает правую часть в формуле Симпсона (3.1) для (3.5) за вычетом слагаемых $f(0)$ и $f(\pi/2)$; $f(x)$ — полная подынтегральная функция в (3.5). После интегрирования и приведения подобных членов получаем

$$F(L) = 4hL^{1-m-n}\left(A_m^{(c)} + \frac{1}{2}\Sigma^{(c)} + A_n^{(c)}\right)/3, \tag{3.7}$$

где

$$A_m^{(c)} = \frac{h^{1-2m}}{4^m}\left(\frac{2+m}{1-m} - 2^{2m+1}\right)\varphi(0); \tag{3.8}$$

$$A_n^{(c)} = \frac{h^{1-2n}}{4^n}\left(\frac{2+n}{1-n} - 2^{2n+1}\right)\varphi(L). \tag{3.9}$$

Отметим, что при $m = n = 0$ $A_m^{(c)} = A_n^{(c)} = 0$; при $m = n = 1/2$ $A_m^{(c)} =$ $= \varphi(0)/2$; $A_n^{(c)} = \varphi(L)/2$.

Аналогично для формулы трапеций

$$F(L) = 2L^{1-m-n}\left[\varphi(L)\int_0^h \vartheta^{1-2n}\,d\vartheta - \frac{h}{2}\varphi(L)h^{1-2n} + \right.$$

$$\left. + \frac{h}{2}\Sigma^{(T)} + \varphi(0)\int_0^h \vartheta^{1-2m}\,d\vartheta - \frac{h}{2}\varphi(0)h^{1-2m}\right], \qquad (3.10)$$

где $\frac{h}{2}\Sigma^{(T)}$ — правая часть (3.2) для (3.5) за вычетом слагаемых $f(0)$ и $f(\pi/2)$; $f(x)$ — полная подынтегральная функция в (3.5). После преобразований получаем

$$F(L) = 2hL^{1-m-n}\left(A_m^T + \frac{1}{2}\Sigma^{(T)} + A_n^{(T)}\right), \qquad (3.11)$$

где

$$A_m^{(T)} = \frac{h^{1-2m}\,m}{2(1-m)}\varphi(0); \qquad (3.12)$$

$$A_n^{(T)} = \frac{h^{1-2n}\,n}{2(1-n)}\varphi(L). \qquad (3.13)$$

Здесь также $A_m^{(T)} = A_n^{(T)} = 0$ при $m = n = 0$ и $A_m^{(T)} = \varphi(0)/2$; $A_n^{(T)} = \varphi(L)/2$ при $m = n = 1/2$.

Формулы (3.7) и (3.11) аналогичны по структуре формулам Симпсона и трапеций соответственно. В них суммы $\Sigma^{(c)}$ и $\Sigma^{(T)}$ такие же, как и суммы по k в (3.1), (3.2), а слагаемые $A_m^{(c)}$, $A_m^{(T)}$ и $A_n^{(c)}$, $A_n^{(T)}$ играют роль $f(0)$ и $f(L)$ в (3.1) и (3.2). Указанные величины, как $f(0)$ и $f(L)$ в программах 33 и 34, могут вводиться в качестве исходных данных. Структура $A_{m,n}^{(c)}$ и $A_{m,n}^{(T)}$ более громоздка, чем $f(0)$ и $f(L)$. Однако она не столь сложна, чтобы вызвать существенные затруднения.

В этом параграфе даны два варианта программ (программы 35 и 36) для общего случая $m \leqslant 1/2$, $n \leqslant 1/2$. Программы основаны на формулах (3.7) и (3.11) соответственно и построены на базе программ 33 и 34. Структура программ аналогична программам § 3.1. В них оставлены свободные ячейки, куда пользователь должен ввести только подпрограмму *регулярной части* подынтегрального выражения в (3.4), т. е. функции $\varphi(x)$. Наличие устранимых особенностей $\varphi(x)$ на концах интервала не играет роли, так как граничные значения $\varphi(0)$ и $\varphi(L)$ фигурируют только в исходных данных.

Очевидно, что длина $\varphi(x)$ получается меньшей, чем длина $f(x)$ в программах 30—34. Поэтому применение метода трапеций приобретает большее значение (см. программы 36, 38, 40).

Приведены также программы для частных случаев, которые распространены в вычислительной практике: $m \leqslant 1/2$, $n = 1/2$ (программы 37, 38); $m = n = 1/2$ (программы 39, 40) и $m = 0$, $n = 1/2$; $n = 0$, $m = 1/2$ (программа 41). В указанных случаях возрастает длина $\varphi(x)$ и скорость счета.

ПРОГРАММА 35

ИНТЕГРАЛ $F(L)=\int\limits_0^L \frac{\varphi(x)\,dx}{x^m (L-x)^n}$ $(m\leqslant 0{,}5,\ n\leqslant 0{,}5)$. ФОРМУЛА (3.7)

	0	1	2	3	4	5
0	$\boxed{\text{В/О}\ 48}$	ПП 68	P $x\rightleftarrows y$ 13	+ 96	P5 51	ПП 68
1	P $x\rightleftarrows y$ 13	БП 58	P0 01	F2 22	↑ 06	F3 32
2	+ 96	P2 21	F4 42	↑ 06	F2 22	P− 83
3	F/−/ 55	× 26				
4				$\varphi(x)$	18 шагов	
5						
6			P8 81	F6 62	↑ 06	F2 22
7	P− 83	x^y 38	↑ 06	F8 82	× 26	P8 81
8	F7 72	↑ 06	F2 22	P+ 93	x^y 38	↑ 06
9	F8 82	× 26	↑ 06	F5 52	+ 96	P5 51

				$\frac{3F(L)\,L^{(m+n-1)}}{4h}$			$\varphi(x)$
Регистры памяти	2	3	4	5	6	7	8
	0	h	L	$F^{(c)}$	$1-2m$	$1-2n$	

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП $F3$. К началу подпрограммы аргумент x функции $\varphi(x)$ введен в регистр $<X>$ и не содержится в других регистрах. Для программирования полностью свободных регистров нет. Можно использовать регистр $<8>$ для хранения промежуточных результатов в подпрограмме $\varphi(x)$ на одном шаге. С этим же условием можно использовать и память стека, если, кроме того, между введением и выборкой чисел из стека не производятся операции ln, e^x, x^y, $\cos x$, $\sin x$, e^{ix}. Результат вычисления $\varphi(x)$ оставляется в регистре $<X>$. При короткой подпрограмме $\varphi(x)$ (см. вводную часть § 3.1) добавить команду БП $F6$.

2. *Вычисление шага* $h=\pi/(3{,}999995\cdot N)$ и *ввод* в $<3>$. Обязательно вводить все цифры числа 3,999995.

3. *Вычисление*

$$F^{(c)}=\frac{\varphi(0)}{4^m}\left(\frac{2+m}{1-m}-2^{2m+1}\right)h^{1-2m}+\frac{\varphi(L)}{4^n}\left(\frac{2+n}{1-n}-2^{2n+1}\right)h^{1-2n}$$

и ввод в <5>. *При грубых расчетах или увеличенном числе шагов* N *в* <5> *можно вводить* 0.

4. *Ввод* 0 в <2>.
5. *Ввод* L в <4>.
6. *Ввод* $(1-2m)$ в <6> и $(1-2n)$ в <7>.
7. *Пуск* В/О С/П . Останов производится по команде переполнения.
8. *Результат* получается умножением содержимого регистра <5> на $4hL^{1-m-n}/3$. Эта операция может быть выполнена командами:

$F6$ ↑ $F7$ + ↑ $F4$ FВП x^y ↑ $F5$ × ↑ $F3$ × 4 × 3 ÷ .

После нажатия соответствующих клавиш на индикаторе окажется искомое значение $F(L)$.

Пример. Вычислить

$$F=\int_0^1 \frac{dx}{(x+2)^{(3/4+4/5)}\, x^{1/4}\, (1-x)^{1/5}}. \tag{3.14}$$

Точное значение [1]:

$$F=\frac{\Gamma(3/4)\cdot\Gamma(4/5)}{3^{3/4}\cdot 2^{4/5}\cdot\Gamma(3/4+4/5)}\approx 0{,}4044084.$$

Для интеграла (3.14) $m=1/4$, $n=1/5$. Подпрограмма $\varphi(x)=1/(x+2)^{(3/4+4/5)}=1/(x+2)^{1{,}55}$ занимает 9 шагов (не считая БП $F6$): 2 + 1 , 5 5 /—/ $\overleftrightarrow{xy}$ x^y БП $F6$. Напомним, что перед началом подпрограммы переменная x находится в регистре <X>.

Результаты вычислений при разном числе шагов N помещены в табл. 1 после программы 36. Для данной $\varphi(x)$ время счета $t\approx N$ мин.

ПРОГРАММА 36

ИНТЕГРАЛ $F(L)=\int_0^L \frac{\varphi(x)\,dx}{x^m(L-x)^n}$ $(m\leqslant 0{,}5,\ n\leqslant 0{,}5)$. ФОРМУЛА (3.11)

	0	1	2	3	4	5
0	P0 01	F4 42	↑ 06	F2 22	P− 83	F/−/ 55
1	× 26					
2			$\varphi(x)$	25 шагов		
3						
4						
5			P8 81	F8 62	↑ 06	F2 22
6	P− 83	x^y 38	↑ 06	F8 82	× 26	P8 81
7	F7 72	↑ 06	F2 22	P+ 93	x^y 38	↑ 06
8	F8 82	× 26	↑ 06	F5 52	+ 96	P5 51
9	F2 22	↑ 06	F3 32	+ 96	PZ 21	БП 58

$$\frac{F(L)\,L^{(m+n-1)}}{h}$$

Регистры памяти	2	3	4	5	6	7	8
	h	h	L	$F^{(T)}$	$(1-2m)$	$(1-2n)$	

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП Р1 . К началу подпрограммы x помещен в регистр $<X>$. Для программирования полностью свободных регистров нет. Регистр $<8>$ можно использовать для хранения промежуточных результатов в подпрограмме $\varphi(x)$ на одном шаге. С этим же условием можно использовать и стек (дополнительные ограничения оговорены в п.1 программы 35). Результат вычисления $\varphi(x)$ оставляется в $<X>$. При короткой подпрограмме $\varphi(x)$ добавляется команда безусловного перехода БП $F5$ (см. вводную часть к § 3.1).

2. *Вычисление шага* $h = \pi/(1{,}999998 \cdot N)$ и *ввод* в $<2>$ и в $<3>$. Во избежание сбоев программы число 1,999998 набирать полностью.

3. *Ввод* $(1-2m)$ в $<6>$ и $(1-2n)$ в $<7>$.

4. *Вычисление* $F^{(T)} = \dfrac{m\varphi(0)}{2(1-m)} h^{1-2m} + \dfrac{n\varphi(L)}{2(1-n)} h^{1-2n}$

и *ввод* в $<5>$. При грубых расчетах или увеличенных N можно в $<5>$ вводить 0.

5. *Ввод* L в $<4>$.

6. *Пуск* В/О С/П . При останове — индукция переполнения.

7. *Результат* получается умножением содержимого регистра $<5>$ на $2hL^{1-m-n}$. Это можно выполнить с помощью команд

$F6$ $\uparrow$ $F7$ $+$ $\uparrow$ $F4$ FВП x^y $\uparrow$ $F5$ $\times$ $\uparrow$ $F3$ $\times$ 2 $\times$,

после которых на индикаторе оказывается искомая величина $F(L)$.

Пример. Вычислить интеграл (3.14). Подпрограмма $\varphi(x)$ совпадает с подпрограммой для этой функции в примере к программе 35, кроме последней команды

Таблица 3

Результаты интегрирования по формулам Симпсона и трапеций при разном числе шагов

Число шагов N	Формула Симпсона: с учетом $F^{(c)}$	Формула Симпсона: без учета $F^{(c)}$	Формула трапеций: с учетом $F^{(T)}$	Формула трапеций: без учета $F^{(T)}$
1	0,3980217	0,3456899	—	—
2	0,4050302	0,3868568	0,3694443	0,2592673
4	0,4044384	0,3981216	0,3931803	0,3549595
8	0,4044048	0,4022073	0,4006026	0,3873312
16	0,4044064	0,4036412	0,4031007	0,3984883
32	—	—	0,4039575	0,4023531

Табличное значение интеграла 0,4044084

$F6$, которую следует заменить на $F5$. В табл. 3 приведены результаты вычислений $F(L)$ при разном числе шагов N. Для данной $\varphi(x)$ время счета $t \approx (N/2)$ мин.

Согласно табл. 3 расчет по формуле Симпсона для достижения одинаковой точности требует существенно меньшего времени, чем по формуле трапеций. Как видно из табл. 3, учет слагаемых $F^{(c)}$ и $F^{(T)}$ ($F^{(c)} = A_m^{(c)} + A_n^{(c)}$; $F^{(T)} = A_m^{(T)} + A_n^{(T)}$ (см. (3.7) и (3.11)) способствует значительному уменьшению погрешностей. Так, при использовании формулы Симпсона и $N = 8$ относительная погрешность снижается с $5 \cdot 10^{-3}$ до $5 \cdot 10^{-6}$, когда в <5> при вводе исходных данных вместо нуля засылается $F^{(c)}$.

ПРОГРАММА 37

ИНТЕГРАЛ $F(L) = \int\limits_0^L \frac{\varphi(x)\,dx}{x^m (L-x)^{1/2}}$ ($m \leqslant 0{,}5$). ФОРМУЛА (3.7)

	0	1	2	3	4	5
0	$\boxed{\text{В/О}\ 48}$	ПП 68	$P\overleftarrow{x y}\!\to$ 13	+ 96	P5 51	ПП 68
1	$P\overleftarrow{x y}\!\to$ 13	БП 58	P0 01	F2 22	↑ 06	F3 32
2	+ 96	P2 21	F4 42	↑ 06	F2 22	P− 83
3	F/−/ 55	× 26				
4						
5			$\varphi(x)$	27 шагов		
6						
7						P8 81
8	F6 62	↑ 06	F2 22	P− 83	x^y 38	↑ 06
9	F8 82	× 26	↑ 06	F5 52	+ 96	P5 51

				$\frac{3F(L)L^{m-1/2}}{4h}$		$\varphi(x)$	
Регистры памяти	2	3	4	5	6	7	8
	0	h	L	$F^{(c)}$	$(1-2m)$		

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП $F3$. К началу подпрограммы $\varphi(x)$ аргумент x этой функции находится в регистре <X>. Для программирования свободен регистр <7>. Кроме того, регистр <8> можно использовать для хранения промежуточных результатов подпрограммы $\varphi(x)$ в течение одного шага. (После прохождения одного шага интегрирования упомянутое число стирается.) Можно также использовать стек (см. ограничения в п.1 программы 35). Р

зультат вычисления $\varphi(x)$ оставляется в регистре <X>. При короткой подпрограмме добавить команду БП FC_x .

2. *Вычисление шага* $h = \pi/(3{,}999995 \cdot N)$ и *ввод* в <3>.
3. *Ввод* $(1-2m)$ в <6>.
4. *Вычисление*

$$F^{(c)} = \frac{\varphi(0)}{4^m}\left(\frac{2+m}{1-m} - 2^{2m+1}\right)h^{1-2m} + \frac{1}{2}\varphi(L) \text{ и ввод в } \langle 5\rangle.$$

5. *Ввод* 0 в <2>.
6. *Ввод* L в <4>.
7. *Пуск* В/О С/П . При останове — индикация переполнения.
8. *Результат* получается умножением содержимого регистра <5> на $4hL^{1/2-m}/3$, что достигается следующей системой команд: $F6$ ↑ $F4$ FБП x^y ↑ $F3$ × 4 × 3 ÷ ↑ $F5$ × . После нажатия соответствующих клавиш на индикаторе оказывается искомая величина $F(L)$.

Примечание. Программа после незначительных корректив может быть использована для вычисления интеграла

$$\Phi(L) = \int_0^L \frac{\varphi(x)\,dx}{x^{1/2}(L-x)^m}, \quad m \leqslant 0{,}5.$$

Для этого в рассматриваемую программу нужно внести следующие изменения:

1. В ячейке 83 команду Р — заменить на Р+ .
2. При вводе исходных данных вместо величины, указанной в п.4. засылать в регистр <5>

$$\Phi^{(c)} = \frac{\varphi(L)}{4^m}\left(\frac{2+m}{1-m} - 2^{2m+1}\right)h^{1-2m} + \frac{1}{2}\varphi(0).$$

Пример. Вычислить

$$F = \int_0^1 \frac{dx}{x^{1/4}(1-x)^{1/2}(1+3x)^{5/4}}. \tag{3.15}$$

Здесь $m = 1/4$.

Точное значение $F = \sqrt{\frac{\pi}{8}}\frac{\Gamma(3/4)}{\Gamma(5/4)} \approx 0{,}8472131$ (см. [2,4]). Длина подпрограммы $\varphi(x) = (1+3x)^{-1{,}25}$ составляет 11 шагов (если не считать БП FC_x): 3 × 1 + 1 , 2 5 /—/ $\overleftarrow{xy}$ x^y БП FC_x . Если при вводе исходных данных занести число (—1,25) в свободный регистр <7>, то подпрограмму можно сократить на три шага: 3 × 1 + ↑ $F7$ $\overleftarrow{xy}$ x^y БП FC_x .

Результаты:

$F = 0{,}8428480$, время счета примерно 1 мин при $N = 2$;

$F = 0{,}8478062$, время счета примерно 2 мин при $N = 4$;

$F = 0{,}8472471$, время счета примерно 4 мин при $N = 8$;
$F = 0{,}8472118$, время счета примерно 9 мин при $N = 16$.
При $N = 16$ относительная погрешность меньше $2\cdot10^{-6}$.

ПРОГРАММА 38

$$\text{ИНТЕГРАЛ } F(L) = \int_0^L \frac{\varphi(x)\,dx}{x^m (L-x)^{1/2}} \; (m \leqslant 0{,}5). \text{ ФОРМУЛА (3.11)}$$

	0	1	2	3	4	5
0	$\boxed{\text{P0 } 01}$	F4 42	↑ 06	F2 22	P− 83	F/−/ 55
1	× 26					
2						
3						
4			φ(x)	34 шага		
5						
6						P8 81
7	F6 62	↑ 06	F2 22	P− 83	x^y 38	↑ 06
8	F8 82	× 26	↑ 06	F5 52	+ 96	P5 51
9	F2 22	↑ 06	F3 32	+ 96	PZ 21	БП 58

				$\frac{F(L)\,L^{m-1/2}}{2h}$		$\varphi(x)$	
Регистры памяти	2	3	4	5	6	7	8
	h	h	L	$F^{(r)}$	$(1-2m)$		

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП P1 . К … чалу подпрограммы аргумент x располагается в регистре $\langle X\rangle$. Д… программирования свободен регистр $\langle 7\rangle$. Для хранения промежут… ных результатов на время прохождения одного шага интегрирова… можно использовать также регистр $\langle 8\rangle$. Можно использовать сте… ограничениями, указанными в п. 1 програмы 35. Результат вычис… ния $\varphi(x)$ оставляется в регистре $\langle X\rangle$. При короткой подпрогра… в ее конце добавляется команда БП FВП.

2. *Вычисление шага* $h = \pi/(1,999998 \cdot N)$ и *ввод* в $<2>$ и в $<3>$. Во избежание сбоев программы недопустимо округлять число 1,999998.

3. *Ввод* $(1 - 2m)$ в $<6>$.

4. *Вычисление*

$$F^{(T)} = \frac{m\varphi(0)}{2(1-m)} h^{1-2m} + \frac{1}{2}\varphi(L) \text{ и } \textit{ввод} \text{ в } \langle 5 \rangle.$$

5. *Ввод* L в $<4>$.

6. *Пуск* В/О С/П. При останове — индикация переполнения.

7. *Результат* получается умножением содержимого регистра $<5>$ на $2hL^{1/2-m}$, что может быть выполнено с помощью следующих команд: $F6$ ↑ $F4$ FВП x^y ↑ $F3$ × 2 × ↑ $F5$ × . После нажатия указанных клавиш на индикаторе оказывается искомая величина $F(L)$.

Примечание. Программа может использоваться после незначительных корректив для вычисления интеграла

$$\Phi(L) = \int\limits_0^L \frac{\varphi(x)\,dx}{x^{1/2}(L-x)^m}, \quad m \leqslant 0,5,$$

в котором подынтегральная функция имеет особенности при $x = L, 0$. Для этого в программу 38 нужно внести следующие изменения:

1. В ячейке 73 команду Р— заменить на Р+ .

2. При вводе исходных данных вместо величины, указанной в п.4, засылать в регистр $<5>$ значение функции

$$\Phi^{(T)} = \frac{m\varphi(L)}{2(1-m)} h^{1-2m} + \frac{1}{2}\varphi(0).$$

Пример. Вычислить интеграл (3.15). Подпрограмма совпадает с подпрограммой для этой функции, которая дана в примере к программе 37, кроме последней команды FC_x, заменяемой на FВП.

Результаты:

$F = 0{,}8240659$, время счета примерно 1 мин при $N = 4$;

$F = 0{,}8398099$, время счета 2 мин при $N = 8$;

$F = 0{,}8446589$, время счета 4 мин при $N = 16$.

Сравнение с результатами вычислений, основанных на формуле (3.7), которая является следствием формулы Симпсона (пример, к программе 37), показывает, что в данном случае достижение одинаковой точности требует намного большего времени счета. Напомним, что исходная формула (3.11) следует из формулы трапеций.

ПРОГРАММА 39

$$\text{ИНТЕГРАЛ } F(L)=\int_0^L \frac{\varphi(x)\,dx}{x^{1/2}(L-x)^{1/2}}.\ \text{ФОРМУЛА (3.7)}$$

	0	1	2	3	4	5
0	$\boxed{\text{В/О 48}}$	ПП 68	P$x\rightleftarrows y$ 13	+ 96	P5 61	ПП 68
1	P$x\rightleftarrows y$ 13	БП 58	P0 01	F2 22	↑ 46	F3 32
2	+ 96	P2 21	P− 83	↑ 06	FВП 65	F4 42
3	× 26	F/−/ 55				
4						
5						
6			φ(x)	36 шагов		
7						
8						
9			↑ 06	F5 52	+ 96	P5 51

				3F/4h	φ(x)		
Регистры памяти	2	3	4	5	6	7	8
	0	h	$\sqrt{L}$	$F^{(c)}$			

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП $F3$. К н чалу подпрограммы аргумент x функции $\varphi(x)$ находится в регист $<X>$. Для программирования свободны регистры $<6>$, $<7>$, <8 Результат вычисления $\varphi(x)$ оставляется в регистре $<X>$. При кор кой подпрограмме добавить в ее конце команду безусловного перехо БП $F9$.

2. *Вычисление шага* $h=\pi/(3{,}999995\cdot N)$ и *ввод* в $<3>$. Во из жание сбоев программы недопустимо округлять число 3,999995.

3. *Вычисление* $F^{(c)}=[\varphi(0)+\varphi(L)]/2$ и *ввод* в $<5>$.

4. *Ввод* 0 в $<2>$.

5. *Ввод* $\sqrt{L}$ в $<4>$.

6. *Пуск* В/О С/П . При останове — индикация переполнен

7. *Результат* получается умножением содержимого регис $<5>$ на $4h/3$, что производится следующей системой кома $F5$ ↑ $F3$ × 4 × 3 ÷ . После нажатия указанных кла на индикаторе появляется искомая величина $F(L)$.

Пример. Вычислить интеграл

$$F=\int_0^3 \frac{dx}{\sqrt{x(3-x)(5-x)(x+1)}}. \tag{3.16}$$

Указанный интеграл выражается через полный эллиптический интеграл первого рода: $F = K(9/10)/\sqrt{5} \approx 1{,}152957$ [2,4].

Длина программы функции $\varphi(x)$: 5 $-$ 6 $+$ $\times$ /−/ FВП F, БП F9 8 шагов (не считая команд БП F9).

Результаты:

$F = 1{,}127316$, время счета примерно 15 с при $N = 1$;
$F = 1{,}150703$, время счета примерно 0,5 мин при $N = 2$;
$F = 1{,}152941$, время счета примерно 1 мин при $N = 4$;
$F = 1{,}152959$, время счета примерно 2 мин при $N = 8$.

Как видно, предельная точность которую может дать микрокалькулятор, достигается при $N = 8$. Величина F отличается от табличной на $2\cdot 10^{-6}$.

ПРОГРАММА 40

$$\text{ИНТЕГРАЛ } F(L)=\int_0^L \frac{\varphi(x)\,dx}{x^{1/2}(L-x)^{1/2}}.\ \text{ФОРМУЛА (3.11)}$$

	0	1	2	3	4	5
0	$\boxed{\text{Р0 01}}$	F2 22	Р− 83	↑ 06	FВП 65	F4 42
1	× 26	F/−/ 55				
2						
3						
4						
5			$\varphi(x)$	42 шага		
6						
7						
8			↑ 06	F5 52	+ 96	Р5 51
9	F2 22	↑ 06	F3 32	+ 96	Р2 21	БП 58

				$\frac{F(L)}{2h}$	$\varphi(x)$		
Регистры памяти	2	3	4	5	6	7	8
	h	h	$\sqrt{L}$	$F^{(T)}$			

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП F1. К началу подпрограммы аргумент x функции $\varphi(x)$ находится в регистре <X>. Для программирования свободны регистры <6>, <7>, <8>.

Результат вычисления $\varphi(x)$ оставляется в регистре $<X>$. При ко-
роткой подпрограмме в ее конце добавляется команда БП $F8$.

2. *Вычисление шага* $h = \pi/(1{,}999998 \cdot N)$ и *ввод* в $<2>$ и $<3>$. Во избежание сбоев программы недопустимо округлять число 1,999998.
3. *Вычисление* $F^{(T)} = [\varphi(0) + \varphi(L)]/2$ и *ввод* в $<5>$.
4. *Ввод* $\sqrt{L}$ в $<4>$.
5. *Пуск* В/О С/П. Останов — индикация переполнения.
6. *Результат* получается умножением содержимого регистра $<5>$ на $2h$, что может быть выполнено следующей системой команд $F5$ ↑ $F3$ × 2 × . После нажатия указанных клавиш на индикаторе появляется искомая величина $F(L)$.

Пример. Вычислить интеграл (3.16). Подпрограмма $\varphi(x)$ в настоящей программе такая же, как и в предыдущей, кроме последней команды $F9$, которую следует заменить на $F8$.

Результаты:

$F = 1{,}153014$, время счета меньше 0,5 мин при $N = 4$;

$F = 1{,}152959$, время счета меньше 1 мин при $N = 8$.

В данном случае предельная точность достигается за то же число шагов, что и в предыдущей программе. Однако время счета оказывается вдвое меньше. Причина такой хорошей сходимости формулы интегрирования (3.11), которая в данном случае совпадает со стандартной формулой трапеций, связана с тем, что отсутствие множителей $(\cos\vartheta)^{1-2m}$, $(\sin\vartheta)^{1-2n}$ при m, $n = 1/2$ (см. формулу (3.5)) приводит к медленному изменению подынтегральной функции, когда $\varphi(x)$ также меняется достаточно плавно.

ПРОГРАММА 41

ИНТЕГРАЛЫ $F_1(L) = \int\limits_0^L \frac{\varphi(x)\,dx}{x^{1/2}}$ и $F_2(L) = \int\limits_0^L \frac{\varphi(x)\,dx}{(L-x)^{1/2}}$. ФОРМУЛА (3.

	0	1	2	3	4	5
0	В/О 48	ПП 68	P$x\rightleftarrows y$ 13	+ 96	Р5 51	ПП 68
1	P$x\rightleftarrows y$ 13	БП 58	Р0 01	F2 22	↑ 06	F3 32
2	+ 96	Р2 21	Р– 83	↑ 06	FВП 65	F4 42
3	× 26	F/–/ 55				
4						
5						
6			$\varphi(x)$	32 шага		
7						
8					↑ 06	F2 22
9	Р+ 93 (Р–) (83)	× 26	↑ 06	F5 52	+ 96	Р5 51

		$\frac{3F(L)}{4h\sqrt{L}}$			$\varphi(x)$		
Регистры памяти	2	3	4	5	6	7	8
	0	h	$\sqrt{L}$	$F^{(c)}$			

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП $F3$. К началу подпрограммы аргумент x функции $\varphi(x)$ находится в регистре <X>. Для программирования $\varphi(x)$ свободны регистры <6>, <7>, <8>. Результат вычисления $\varphi(x)$ оставляется в <X>. При короткой подпрограмме следует добавить в ее команду БП 8 . *Для вычисления* $F_1(L)$ в ячейку 90 программы помещается команда Р+. *Для вычисления* $F_2(L)$ в ту же ячейку вместо Р+ помещается команда Р—.
2. *Вычисление* $h = \pi/(3{,}999995 \cdot N)$ *и ввод в* <3>.
3. *Вычисление* $F^{(c)}$ *и ввод* в <5>:

$$F^{(c)} = \begin{cases} \varphi(0)/2 \text{ для } F_1(L); \\ \varphi(L)/2 \text{ для } F_2(L). \end{cases}$$

4. *Ввод* $\sqrt{L}$ в <4>.
5. *Ввод* 0 в <2>.
6. *Пуск* В/О С/П. Останов — индикация переполнения.
7. *Результат* получается умножением содержимого регистра <5> на $4h\sqrt{L}/3$, что может быть выполнено следующей системой команд:

$$F5 \quad \uparrow \quad F4 \quad \times \quad \uparrow \quad F3 \times 4 \quad \times \quad 3 \quad \div \quad .$$

После нажатия указанных здесь клавиш на индикаторе появится искомая величина $F_1(L)$ или $F_2(L)$.

Примеры:

1. Вычислить интеграл $F_1 = \int\limits_0^1 \frac{1+2x}{\sqrt{x}}\,dx$.

Табличное значение (2) $F_1 = 10/3 = 3{,}33333...$ Подпрограмма $\varphi(x) = 1 + 2x$ (2 × 1 + БП 8) содержит 4 шага, не считая команды БП 8 . Вводим в ячейку 90 команду Р+, а в регистр <5> $\varphi(0)/2 = 1/2$. При числе шагов $N = 10$ (время счета примерно 3 мин) $F_1 = 3{,}333343$. Относительная погрешность примерно $3 \cdot 10^{-5}$.

2. Вычислить интеграл $F_2 = \int\limits_0^2 \frac{1+2x}{\sqrt{2-x}}\,dx$.

Табличное значение $F_2 = 22\sqrt{2}/3 \approx 10{,}370898$. Подпрограмма $\varphi(x)$ такая же, как и в примере 1. Вводим в ячейку 90 команду Р—. В регистр <5> вводим $\varphi(2)/2 = 2{,}5$. В <4> вводим $\sqrt{2}$. При числе шагов $N = 10$ $F_2 = 10{,}370884$. Время счета примерно 3 мин. Относительная погрешность примерно $1 \cdot 10^{-6}$.

3.3. ГЛАВНОЕ ЗНАЧЕНИЕ ИНТЕГРАЛА ПО КОШИ

ОПРЕДЕЛЕНИЕ

Главным значением интеграла по Коши называется предел (см., например, [5]):

$$F = VP \int_{x_0}^{L} f(x)\,dx = \lim_{\varepsilon \to 0} \left[\int_{x_0}^{x_P - \varepsilon} f(x)\,dx + \int_{x_P+\varepsilon}^{L} f(x)\,dx \right] \quad (3.17)$$

(x_P — особая точка $f(x)$).

Далее рассматривается частный случай (3.17), когда

$$f(x) = \varphi(x)/(x - x_P), \quad (3.18)$$

где $\varphi(x)$ — регулярная функция за исключением возможно устранимых разрывов в отдельных точках интервала (x_0, L) и на его границах. Интегралы (3.17) с функцией (3.18) часто встречаются в приложениях математической физики.

МЕТОД РАСЧЕТА

Ограничимся случаем, когда отношение длин $(x_P - x_0)$ и $(L - x_P)$ равно отношению натуральных чисел:

$$(L - x_P)/(x_P - x_0) = m/n. \quad (3.19)$$

При этом условии всегда можно выбрать так размер шага h, чтобы в отрезках $(x_P - x_0)$ и $(L - x_P)$ укладывалось четное число шагов.

К каждому слагаемому в (3.17) применим формулу Симпсона (3.1 с учетом (3.18):

$$F = \lim_{\varepsilon \to 0} \left[\int_{x_0}^{x_P - \varepsilon} \frac{\varphi(x)\,dx}{x - x_P} + \int_{x_P+\varepsilon}^{L} \frac{\varphi(x)\,dx}{x - x_P} \right] = \frac{h}{3} \lim_{\varepsilon \to 0} \left[f(x_0) + \sum_{k=0}^{N_1 - 1} \left\{ 4f[x_0 + (2k+1)h] + 2\varkappa f(x_0 + 2kh) \right\} + f(x_P - \varepsilon) + f(x_P + \varepsilon) + \sum_{k=0}^{N_2+1} \{4f[x_P + (2k+1)h] + 2\varkappa f(x_P + 2kh)\} + f(L) \right], \quad (3.20)$$

где $f(x) = \varphi(x)/(x - x_P)$; $\varkappa = 0$; 1 при $k = 0$ и $k > 0$ соответственно.

Величина h в (3.20), строго говоря, зависит от ε, если заданы числа двойных шагов N_1 и N_2 на соответствующих интервалах. Однако будем полагать h константой, связанной с полным числом шагов соотношением

$$h = (L - x_0)/(2N). \quad (3.21)$$

Легко показать, что это не сказывается на результатах. Нужно только, чтобы в каждом отрезке $(x_P - x_0)$ и $(L - x_P)$ укладывалось целое число двойных шагов. Если m и n в (3.19) не взаимно простые, то минимальное число двойных шагов на полном интервале $N_{\min} = (m + n)/D$, где D — общий наибольший делитель m и n. При этом максимальн значение шага определяется (3.21) при $N = N_{min}$. Большие N должны быть кратны N_{min}.

Таким образом, предельный переход в (3.20) сводится к нахождению

$$\lim_{\varepsilon \to 0} [f(x_P - \varepsilon) + f(x_P + \varepsilon)] = \lim_{\varepsilon \to 0} \left[\frac{\varphi(x_P - \varepsilon)}{-\varepsilon} + \frac{\varphi(x_P + \varepsilon)}{\varepsilon} \right] = 2\varphi'(x_P). \tag{3.22}$$

Подставляя (3.22) в (3.20), получаем следующий вариант формулы Симпсона для вычисления главного значения интеграла по Коши:

$$F = \frac{2h}{3} \left\{ \varphi'(x_P) + \frac{\varphi(x_0)}{2(x_0 - x_P)} + \frac{\varphi(L)}{2(L - x_P)} + \sum_{k=0}^{N-1} (2f[x_0 + (2k+1)h] + \tau f(x_0 + 2kh)) \right\}. \tag{3.23}$$

Здесь $\tau = 1$ при всех k, кроме $k = 0$ и того k, где $x_0 + 2kh = x_P$. В этих двух точках $\tau = 0$. Отличие (3.23) от стандартной формулы Симпсона состоит в присутствии слагаемого $\varphi'(x_P)$ и в том, что в сумме по k должен быть пропущен член, соответствующий $x = x_P$. В приведенной далее программе 42 сумма $F^{(c)} = \varphi'(x_P) + \varphi(x_0)/(2(x_0 - x_P)) + \varphi(L)/(2(L - x_P))$ вычисляется отдельно и вводится в качестве исходного данного.

ПРОГРАММА 42

Главное значение интеграла по Коши $F = \mathrm{VP} \int_{x_0}^{L} \frac{\varphi(x)\,dx}{x - x_P}$ (отношение $(L - x_P)/(x_P - x_0) = m/n$, где m, n — натуральные числа). Вариант (3.23) формулы Симпсона

	0	1	2	3	4	5
0	В/0 48	ПП 68	F2 22	+ 96	Р5 51	F4 42
1	↑ 06	F2 22	− 86	FВП 65	ПП 68	F2 22
2	БП 58	Р0 01	F2 22	↑ 06	F3 32	+ 96
3	Р2 21	↑ 06	F6 62	− 86	Р7 71	↑ 06
4	F3 32	÷ 36	2 24	× 26	F/−/ 55	1 14
5	− 86	РВ/0 49	Р0 01			
6						
7			φ(x)	20 шагов		
8						↑ 06
9	F7 72	÷ 36	↑ 06	F5 52	+ 96	Р5 51

	$\frac{3F}{2h}$						$\varphi(x)$
Регистры памяти	2	3	4	5	6	7	8
	x_0	h	$L-2h$	$F^{(c)}$	x_p		

Порядок работы

1. *Программирование* $\varphi(x)$. Начало подпрограммы БП Р/—/ . Аргумент x функции $\varphi(x)$ содержится в регистре <2>. Для программирования $\varphi(x)$ свободен регистр <8>. Можно также использовать стек, если между засылкой величин в стек и их выборкой из стека не производятся операции $\ln x$, e^x, x^y, $\cos x$, $\sin x$, e^{ix}. Результат вычисления $\varphi(x)$ оставляется в <X>. При короткой подпрограмме следует добавить в ее конце БП F— .

2. *Вычисление шага* $h = (L - x_0)/(2N)$ и ввод в <3>. *Число* N *должно быть кратно* $(m + n)/D$, где m и n — натуральные числа, упомянутые в заголовке программы, а D—их общий наибольший делитель.

3. *Вычисление* $(L - 2h)$ *и ввод* в <4>.

4. *Вычисление* $F^{(c)} = \varphi'(x_P) + \frac{1}{2}\left[\frac{\varphi(x_0)}{x_0 - x_P} + \frac{\varphi(L)}{L - x_P}\right]$ и *ввод* в <5>.

5. *Ввод* x_0 в <2>.

6. *Ввод* x_P в <6>.

7. *Пуск* В/О С/П . Останов—индикация переполнения.

8. *Результат* получается умножением содержимого регистра <5> на $2h/3$, что производится следующей системой команд: $F5$ ↑ $F3$ × 2 × 3 ÷ . После нажатия указанных клавиш на индикаторе должна появиться искомая величина F.

Пример. Вычислить главное значение интеграла $F = -\mathrm{VP}\int_{-1}^{\infty}\frac{e^{-x}}{x}dx$ (интегральная показательная функция Ei (1) — см. § 2.3).

Здесь $x_0 = -1$; $x_p = 0$. Выберем в качестве верхнего предела $L = 11$, для которого модуль подынтегральной функции — e^{-x}/x достаточно мал ($\sim 1\cdot 10^{-6}$). Отношение $(L - x_0)/(x_p - x_0) = 11$. Наименьшее число двойных шагов $N_{\min} = 11 + 1 = 12$. Длина подпрограммы функции $\varphi(x) = -e^{-x}$ составляет 4 шага, не считая команды БП F— : $F2$ /—/ Р÷ /—/ БП F— . Величина $F^{(c)}$, вводимая в <5>, равна: $1 + (e - e^{-11}/11)/2 = 2{,}3591399$.

Результаты:
$F = 1{,}8938984$, время счета примерно 3,5 мин при $N = 12$;
$F = 1{,}8950173$, время счета примерно 7 мин при $N = 24$;
$F = 1{,}8950963$, время счета примерно 10 мин при $N = 36$.

Табличное значение [4] Ei (1) = 1,89511781 ... Относительная погрешность при $N = 36$ составляет примерно $1\cdot 10^{-5}$.

3.4. ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТОВ РЯДА ФУРЬЕ И ЧАСТИЧНЫХ СУММ РЯДА ФУРЬЕ

Ряд Фурье, порожденный функцией $f(x)$, имеет вид

$$f(x) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{2\pi n}{L}x\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{2\pi n}{L}x\right). \qquad (3.24)$$

Коэффициенты ряда

$$A_n = \frac{2}{L\varepsilon_n} \int_0^L f(x) \cos\left(\frac{2\pi n}{L} x\right) dx; \quad B_n = \frac{2}{L} \int_0^L f(x) \sin\left(\frac{2\pi n}{L} x\right) dx, \tag{3.25}$$

где

$$\varepsilon_n = \begin{cases} 2, & n = 0; \\ 1, & n < 0. \end{cases}$$

Коэффициенты A_n и B_n в программе 43 вычисляются численным интегрированием (3.25) по формуле Симпсона (на базе программы 33). Частичные суммы ряда Фурье

$$S_N = \sum_{n=0}^{N} A_n \cos\left(\frac{2\pi n}{L} x\right) + \sum_{n=1}^{N} B_n \sin\left(\frac{2\pi n}{L} x\right) \tag{3.26}$$

вычисляются как непосредственным суммированием, так и по программе, содержащей в качестве подпрограммы алгоритм вычисления коэффициентов A_n и B_n.

В первом случае на каждом шаге вводится очередная пара коэффициентов A_n и B_n. После этого производится пуск и при последующем счете к частичной сумме прибавляется слагаемое $A_n \cos(2\pi nx/L) + B_n \sin(2\pi nx/L)$ и подготавливаются данные для следующего пуска и т. д. В этом варианте основное время затрачивается на набор и введение в память калькулятора величин A_n и B_n, даже если они известны.

Во втором случае, когда вычисление A_n и B_n может быть выполнено по удобному алгоритму, для них набирается соответствующая программа и расчет S_N с заданным N выполняется автоматически.

ПРОГРАММА 43

КОЭФФИЦИЕНТЫ РЯДА ФУРЬЕ A_n и B_n. ФОРМУЛЫ (3.25)

	0	1	2	3	4	5
0	$\boxed{\text{В/О}\ 48}$	ПП 68	F2 22	+ 96	↑ 06	F2 22
1	FВП 65	$\overrightarrow{xy}$ 16	Р5 51	ПП 68	F2 22	Р5 51
2	БП 58	Р0 01	F2 22	↑ 06	F3 32	− 86
3	Р2 21	↑ 06	F4 42	× 26	Р− 83 (Р+) (93)	Р6 61
4						
5						
6			f(x)	30 шагов		
7						
8						
9	↑ 06	F6 62	× 26	↑ 06	F5 52	+ 96

				$\frac{3L}{4h}$ A_n; B_n		$f(x)$	
Регистры памяти	2	3	4	5	6	7	8
	L	h	$\frac{2\pi n}{L}$	$F^{(0)}$			

Порядок работы

а) Вычисление косинус-коэффициентов A_n

1. *Программирование* $f(x)$. Начало подпрограммы БП $\div$. Ар-гумент x функции $f(x)$ содержится в <2>. Для программировани свободны регистры <7> и <8>. Результат вычисления $f(x)$ оставля-ется в <X>. При короткой подпрограмме $f(x)$ следует добавить в е конце команду безусловного перехода БП — .

2. *Вычисление шага интегрирования* $h = L/(6(n+3))$ и *ввод* <3>.

3. *Вычисление* $F^{(0)} = [f(+0) + f(L-0)]/2$ и *ввод* в <5>.

4. *Вычисление* $2\pi n/L$ и *ввод* в <4>.

5. *Ввод* L в <2>. *Пуск* В/О С/П . Останов — по переполнени

6. *Результат* получается умножением содержимого регистра <5 на $4h/(3L) = 2/(9(n+3))$, что производится с помощью следующи команд: $F5$ ↑ $F3$ × 4 × 3 ÷ ↑ $\widehat{L}$ ÷ . Здесь $\widehat{L}$ означа набор числа L на клавиатуре. После нажатия указанных клавиш н индикаторе оказывается искомое значение A_n. *При вычислении* A_0 *р зультат следует дополнительно разделить на 2.*

б) Вычисление синус-коэффициентов B_n

Это вычисление требует внесения в предыдущую программу сл дующих изменений:

— в ячейку 34 программы вместо команды Р— следует помести команду Р+ (см. программу);

— при вводе исходных данных вместо $F^{(0)} = \frac{1}{2}[f(+0) + f(L-0)]$ (п. 3) следует ввести $F^{(0)} = 0$ в регистр <5 В остальном порядок действий такой же, как и в п. а.

Пример. Вычислить A_8 и B_8 для функции $f(x) = x$ и $0 \leqslant x \leqslant 1$ (треуго ный периодический сигнал). Точные значения A_n и B_n: $A_0 = 0{,}5$; $A_n = 0$ ($n >$ $B_n = -1/(n\pi)$.

Подпрограмма $f(x)$, если не считать команду БП— содержит только од команду: $F2$ (вызов x из <2> в <X>).

Результаты:

$A_8 = 8{,}848484 \cdot 10^{-9}$, точное значение $A_8 = 0$;

$B_8 = -3{,}986852 \cdot 10^{-2}$, точное значение $B_8 = -1/(8\pi) \approx -3{,}978873 \cdot 10$

Время счета примерно 7 мин (на каждый из коэффициентов).

ПРОГРАММА 44

ЧАСТИЧНЫЕ СУММЫ РЯДА ФУРЬЕ. ФОРМУЛЫ (3.26)

	0	1	2	3	4	5
0	С/П 78	F6 62	1 14	+ 96	P6 61	F2 22
1	× 26	P↑ 03	P, 43	F4 42	× 26	↑ 06
2	F5 52	+ 96	P5 51	P/−/ 53	↑ 06	F3 32
3	× 26	↑ 06	F5 52	+ 96	P5 51	С/П 78
4						
5						
6			A_n и B_n	31 шаг		
7						
8						
9	F7 72	↑ 06	F6 62	− 86	РПП 69	Р0 01

		A_n	B_n	S_N	n		A_n B_n
Регистры памяти	2	3	4	5	6	7	8
	$\frac{2\pi x}{L}$	A_1	B_1	A_0	1	N	

Порядок работы

а) Отдельный ввод коэффициентов A_n и B_n

1. *Вычисление* $2\pi x/L$ и *ввод* в <2>.
2. *Ввод коэффициентов Фурье* A_0, A_1, B_1 соответственно в регистры <5>, <3>, <4>.
3. *Ввод* 1 в <6>.
4. Пуск В/О С/П .
5. *Результат*: После останова в регистре <5> оказывается первая частичная сумма $S_1 = A_0 + A_1 \cos(2\pi x/L) + B_1 \sin(2\pi x/L)$.
6. *Для получения следующей частичной суммы* A_2 вводится в <3> и B_2 в <4>, а затем осуществляется повторный пуск В/О С/П . После останова в регистре <5> S_2. Точно так же получаются и последующие S_N. Номер n очередных A_n и B_n подготавливается на предыдущем цикле и содержится в <6>.
7. Если необходимо суммировать ряд при другом x, то вводится соответствующее значение $2\pi x/L$, в <2>, 1 в регистр <6> и коэффициенты A_0, A_1, B_1 соответственно в <5>, <3>, <4>. Дальнейшие действия производятся в указанном выше порядке.

б) Вычисление A_n и B_n в программе. Автоматический счет частичных сумм

Если существует достаточно простой алгоритм расчета A_n и B_n (30 шагов на соответствующую подпрограмму), то можно использовать другой вариант, при котором A_n и B_n вычисляются внутри программы и частичная сумма S_N с заданным N рассчитывается автоматически. В программной памяти калькулятора выделена область (см. программу), куда должна вводиться подпрограмма для вычисления A_n и B_n как функций порядкового номера членов ряда n.

1. *Программирование* A_n *и* B_n. Начало подпрограммы БП F ÷ . Номер члена ряда n берется из <6>. Для программирования свободен регистр <8>. Можно использовать и регистры стека для хранения промежуточных данных на одном цикле по n (ограничения см. в программе 35). Результаты вычисления A_n и B_n вводятся соответственно в <3> и <4>. *Если коэффициенты* A_n *или* B_n *не меняются, то программировать их не следует*, так как значения A_1 и B_1 помещены в <3> и <4> при вводе исходных данных (см. ниже п. 3). При «короткой» подпрограмме добавить в ее конце команду безусловного перехода БП — .

Обращаем внимание читателя на то, что при вводе подпрограммы *стирается команда* останова С/П в ячейке 35. *Эта команда должна быть восстановлена при работе по варианту* а).

2. *Вычисление* $2\pi x/L$ *и ввод в* <2>.
3. *Ввод коэффициентов* A_0, A_1, B_1 соответственно в регистры <5>, <3>, <4>.
4. *Ввод номера* частичной суммы N в <7>.
5. *Ввод* 1 в <6>.
6. *Пуск* В/О С/П .
7. *Результат*: После останова в регистре <5> требуемая частичная сумма S_N.
8. *Если нужно вычислить новую частичную сумму с* $N' > N$, достаточно в <7> ввести новое значение N' и осуществить пуск В/О С/П .
9. При расчете S_N с другим x все пункты, кроме п. 1, следует повторить.

Пример. Рассчитать частичные суммы ряда Фурье, изображающего треугольный периодический сигнал (пример к программе 43). Коэффициенты ряда $A_0 = 0{,}5$; $A_n = 0$ $(n > 0)$; $B_n = -1/n\pi$.

Согласно п. 1 варианта б) достаточно программировать B_n, так как для всех $n \geqslant 1$ $A_n = A_1 = 0$. Подпрограмма B_n (F6 ↑ Р× × /—/ F, Р4 БП —) содержит 7 шагов, не считая команды безусловного перехода БП — .

Вводим 0,5 в <5>; 0 в <3>, $-1/\pi$ в <4>, 1 в <6>. Принимаем $L = 1$, $x = 0{,}3$ и вводим $2\pi x/L = 1{,}884955$ в <2>.

Результаты:

$S_4 = 0{,}2775015$; $S_{50} = 0{,}3023119$;
$S_{16} = 0{,}3112375$; $S_{100} = 0{,}3011562$;
$S_{30} = 0{,}3038511$; $S_{150} = 0{,}3007709$.

Время счета примерно $(N/6)$ мин.

Как видно, сходимость ряда к значению функции $f(x) = x = 0{,}3$ весьма медленная.

3.5. ИНТЕГРИРОВАНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПЕРВОГО И ВТОРОГО ПОРЯДКА

1. Численное интегрирование дифференциального уравнения первого порядка

$$\frac{dy}{dx}=f(x,y) \tag{3.27}$$

при граничном условии

$$y(x_0)=y_0$$

проводится ниже методом Рунге—Кутта четвертого порядка и исправленным методом Эйлера (см., например, [10]).

а) Метод Рунге—Кутта. Расчетная схема имеет вид

$$\left.\begin{aligned}
&k_1=\frac{h}{2}f(x_n,y_n);\; k_2=\frac{h}{2}f\left(x_n+\frac{h}{2},\,y_n+k_1\right);\\
&k_3=\frac{h}{2}f\left(x_n+\frac{h}{2},\,y_n+k_2\right);\; k_4=\frac{h}{2}f(x_n+h,y_n+2k_3);\\
&y_{n+1}=y_n+(k_1+2k_2+2k_3+k_4)/3.
\end{aligned}\right\} \tag{3.28}$$

Применение указанных формул позволяет по значению функции $y(x)$ в n-м узле определить ее значение в $(n+1)$-м узле разностной сетки. Схема (3.28) реализована в программе 45, которая взята из книги Я. К. Трохименко и Ф. Д. Любича [1]. После каждого пуска программы определяется значение функции в точке, отстоящей от начальной на один шаг. Величина последнего находится по формуле

$$h=(L-x_0)/N, \tag{3.29}$$

где L — конечное значение x; N — число шагов на интервале (x_0, L). Когда величина шага выбрана, интегрирование можно вести и при числе шагов, большем N, что даст значения $y(x)$ при $x>L$ (см. пример к программе 45).

Оценку погрешности можно произвести путем использования половинного шага. Поскольку схема (3.28) имеет четвертый порядок точности, погрешность уменьшается примерно в 16 раз при удвоении N. Отметим, что чрезмерное уменьшение шагов может привести к возрастанию погрешностей за счет ошибок округления (см., например, [3]).

б) Исправленный метод Эйлера [10]:

$$\begin{aligned}
&y^*_{n+1}=y_n+hf(x_n,y_n);\\
&y_{n+1}=y_n+h\,[f(x_n,y_n)+f(x_n+h,y^*_{n+1})]/2.
\end{aligned} \tag{3.30}$$

Комбинация формул (3.30) может рассматриваться как вариант метода Рунге—Кутта [11] и как простейший вариант метода прогноза-коррекции [12]. Первая из формул, являющаяся формулой Эйлера, дает прогноз, а вторая — коррекцию. В отличие от классических вариантов метода прогноза-коррекции схема (3.30) является одноступенчатой.

Сложность численной реализации схем (3.28) и (3.30) неодинакова: при использовании (3.28) объем программной памяти, который удается выделить на программирование правой части дифференциального уравнения (3.27), т. е. функции $f(x, y)$ составляет 24 шага, тогда как для (3.30) максимальная «длина» $f(x, y)$ достигает 34 шагов. Это позволяет вести счет автоматически «очередями» по K шагов. Длина $f(x, y)$ в этом случае снижается до 28 шагов. Число K может быть задано в каждой «очереди».

Отметим, что при использовании (3.30) на каждом шаге интегрирования требуется два обращения к подпрограмме правой части, тогда как при работе с (3.28) необходимы четыре обращения.

2. Интегрирование дифференциального уравнения второго порядка

$$d^2y/dx^2 = f(x, y, y') \tag{3.31}$$

с начальными условиями Коши

$$y(x_0) = y_0;\; y'(x_0) = y_0' \tag{3.32}$$

реализовано на основе исправленного метода Эйлера, аналогичного схеме (3.30). Уравнение (3.31) и начальные условия (3.32) эквивалентны системе уравнений первого порядка

$$y' = z;\; z' = f(x, y, z) \tag{3.33}$$

с начальными условиями

$$y(x_0) = y_0;\; z(x_0) = y_0'. \tag{3.34}$$

Исправленный метод Эйлера для (3.33) характеризуется следующей схемой:

$$\begin{aligned}
y_{n+1}^* &= y_n + hz_n;\\
z_{n+1}^* &= z_n + hf(x_n, y_n, z_n);\\
y_{n+1} &= y_n + h(z_n + z_{n+1}^*)/2;\\
z_{n+1} &= z_n + h[f(x_n, y_n, z_n) + f(x_n + h, y_{n+1}^*, z_{n+1}^*)]/2.
\end{aligned} \tag{3.35}$$

Систему (3.35) удобно для программирования преобразовать к виду

$$\left.\begin{aligned}
z_{n+1}^* &= z_n + hf(x_n, y_n, z_n);\\
y_{n+1} &= y_n + h(z_n + z_{n+1}^*)/2;\\
z_{n+1} &= [z_n + z_{n+1}^* + hf(x_n + h, y_{n+1}, z_{n+1}^*)]/2.
\end{aligned}\right\} \tag{3.36}$$

Метод реализован в программе 47. Максимальная длина $f(x, y, y')$ составляет 25 шагов. На каждом шаге интегрирования производится два обращения к подпрограмме $f(x, y, y')$.

ПРОГРАММА 45

ИНТЕГРИРОВАНИЕ ОБЫКНОВЕННОГО ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ (3.27). МЕТОД РУНГЕ — КУТТА ЧЕТВЕРТОГО ПОРЯДКА. ФОРМУЛЫ (3.28)

	0	1	2	3	4	5
0	$\boxed{\text{В/0 } 48}$	ПП 68	F4 42	ПП 68	Р÷ 33	+ 96
1	Р5 51	ПП 68	F4 42	+ 96	Р5 51	F4 42
2	+ 96	Р4 41	ПП 68	Р÷ 33	3 34	÷ 36
3	Р3 31	Р4 41	С/П 78	F2 22	↑ 06	F6 62
4	+ 96	Р2 21				
5						
6			$f(x,y)$	24 шага		
7						
8			↑ 06	F6 62	× 26	↑ 06
9	F3 32	+ 96	Р4 41	F5 52	+ 96	Р5 51

	x_n	y_n	y_n			$f(x, y)$	
Регистры памяти	2	3	4	5	6	7	8
	x_0	y_0	y_0	$3y_0$	$h/2$		

Порядок работы

1. *Программирование правой части уравнения* (3.27) — функции $f(x, y)$. Начало подпрограммы — БП F4 . Переменные x и y следует брать соответственно из регистров <2> и <4>. Для программирования свободны регистры <7> и <8>. Результат вычисления $f(x, y)$ оставляется в регистре <X>. При «короткой» программе $f(x, y)$ (см. с. 67) в ее конце добавляется команда безусловного перехода БП F8 .
2. *Вычисление половинного шага* $h/2 = (L - x_0)/(2N)$ и *ввод* в <6>.
3. *Ввод* x_0 в <2>.
4. *Ввод* y_0 в <3> и в <4>.
5. *Ввод* $3y_0$ в <5>.
6. *Пуск* В/О С/П .
7. *Результат*: После останова в регистрах <3> и <4>, а также на индикаторе оказывается значение функции в первом узле: $y(x + h)$; в регистре <2> — значение аргумента: $x = x + h$. Повторный пуск В/О С/П дает значения функции и аргумента в следую-

шем узле и т. д. Естественно, что указанный процесс можно пр
должить и за предварительно выбранную границу интервала $x=$
Если желательно вернуться назад на некоторое число шагов, т
в регистре <6> ($h/2$) следует заменить на ($-h/2$) и затем пр
извести нужное число пусков, т. е. одношаговых смещений по ра
ностной сетке.

Пример. Проинтегрировать дифференциальное уравнение

$$dy/dx=(1+x)^2-y \qquad (3.37)$$

при граничном условии $(y)_{x=0}=1$.

Точное решение уравнения при указанном граничном условии имеет ви

$$y=1+x^2. \qquad (3.38)$$

Длина программы правой части уравнения $f(x,\ y)=(1+x)^2-y$ ($F2$ 1 $F/-/$ ↑ $F4$ — БП $F8$) 7 шагов без команды БП $F8$.

Результаты. Ниже приведены вычисленные значения y (2,5) при $N=$ 10; 20 (соответственно величина шага 0,5; 0,25 и 0,125):

y (2,5) = 7,251520 при $N=5$;
y (2,5) = 7,250085 при $N=10$;
y (2,5) = 7,250004 при $N=20$.

Сопоставление приведенных данных и учет того, что при удвоении N п грешность должна уменьшаться более чем на порядок, позволяют заключить, ч при $N=16$ погрешность не превышает $5\cdot10^{-6}$, что соответствует относительн погрешности $7\cdot10^{-7}$. Точное значение y (2,5) = 7,25.

Время счета $t\approx(N/3)$ мин, где N — число шагов.

ПРОГРАММА 46

ИНТЕГРИРОВАНИЕ ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ $dy/dx=f(x,\ y)$. ИСПРАВЛЕННЫЙ МЕТОД ЭЙЛЕРА. ФОРМУЛЫ (3.30)

	0	1	2	3	4	5
0	В/0 40	ПП 68	4 44	F3 32	XY 16	+ 96
1	Р3 31	+ 96	Р5 51	F4 42	↑ 06	F2 22
2	+ 96	Р2 21	ПП 68	4 44	F5 52	+ 96
3	2 24	÷ 36	Р3 31	F6 62	1 14	− 86
4	Р6 61	РПП 59	Р0 01	С/П 78		
5						
6			$f(x,y)$	28 шагов		
7						
8						
9			↑ 06	F4 42	× 26	↑ 06

	x_h	y_h				$f(x, y)$	
Регистры памяти	2	3	4	5	6	7	8
	x_0	y_0	h		$K-1$		

Порядок работы

1. *Программирование* $f(x, y)$ — правой части (3.27). Начало подпрограммы БП 4 . Переменные x и y берутся из регистров <2> и <3>. Для программирования $f(x, y)$ свободны регистры <7> и <8>. Результат вычисления $f(x, y)$ оставляется в регистре <X>. При «короткой» подпрограмме $f(x, y)$ (см. с. 67) в ее конце добавляется команда безусловного перехода БП $F9$.
2. *Ввод шага* h в <4>.
3. *Ввод* x_0 в <2>.
4. *Ввод* y_0 в <3>.
5. *Ввод* $(K - 1)$ в <6>. K — число шагов автоматического счета.
6. *Пуск* В/О С/П .
7. *Результат*: После останова в регистре <3> оказывается приближенное значение функции на K-м шаге, т. е. величина $y_k = y(x_0 + Kh)$.
8. *Повторный пуск*. В регистр <6> вводится новое число $(K' - 1)$ и затем осуществляется пуск В/О С/П . Если желателен пошаговый счет, то в <6> вводится любое число, меньшее 1, и далее нажимаются клавиши В/О С/П перед очередным шагом. После останова в регистре <3> оказывается значение функции на соответствующем шаге.
9. *Одношаговая программа*. Увеличение длины подпрограммы $f(x, y)$ до 34 шагов достигается переходом к одношаговому варианту программы. Для этого в данной программе следует произвести следующие изменения:

а) заменить команду «4» в ячейках 02 и 23 на команду «3»;

б) заменить команду $F6$ в ячейке 33 на команду С/П;

в) убрать команды из ячеек 34—43.

В получившемся варианте программы ввод подпрограммы $f(x, y)$ должен начинаться с ячейки 34 (БП 3). Длина подпрограммы увеличивается до 34 шагов. Для программирования $f(x, y)$ свободны регистры памяти <6>, <7>, <8>. При вводе исходных данных п. 5 следует опустить. В остальном порядок работы такой же, как и при использовании основного варианта программы.

Пример.

1. Уравнение $y' = y$ с начальным условием $y(0) = 1$.
Точное решение: $y_T = e^x$.
Результаты интегрирования ($h = 0{,}05$):

x	0	0,5	1	1,5	2	2,5	3
y	1	1,64839	2,71719	4,47899	7,38312	12,1703	20,0613
y_T	1	1,64872	2,71828	4,41869	7,38905	12,1825	20,0851

Счет проводился циклами («очередями») по 10 шагов. Время одного цикла около 1 мин. Общее время интегрирования на интервале (0; 3) 8 мин.

2. Уравнение Риккати

$$x^3y' - x^6y^2 - x^2(2x-3)y + 3 = 0. \qquad (3.39)$$

с начальным условием $y(-1) = (3 - e^4)/(1 + e^4) = -0{,}9280551$. Точное решение этого уравнения с указанным начальным условием имеет вид [13]

$$y_T = (1 - 3e^{4x})/\left(x^3(1 + e^{4x})\right). \qquad (3.40)$$

Длина подпрограммы правой части уравнения (3.39), разрешенного относительно производной, имеет длину 28 шагов:

F2 ↑ F/—/ × 3 ÷ P7 F3 F/—/ × P8 F2

3 ÷ F, 2 — ↑ F3 × /—/ ↑ F7 F,

— ↑ F8 +

Результаты интегрирования ($h = 0{,}01$):

x	−0,75	−0,5	−0,4	−0,3	−0,2	−0,1	−0,05
y	−1,9195	−4,1719	−5,0907	−2,6490	30,216	597,66	6060,8
y_T	−1,9207	−4,1855	−5,1261	−2,7444	30,013	605,25	6405,3

Общее время счета на интервале $-1 \leqslant x \leqslant -0{,}05$ (95 шагов) около 30 мин.

Погрешность при $x = -0{,}05$ достигает 6%. Это, по-видимому, немного, если учесть, что в близко расположенной точке $x = 0$ функция (3.40) имеет особенность вида $1/x^3$.

ПРОГРАММА 47

ИНТЕГРИРОВАНИЕ ДИФФЕРЕНЦИАЛЬНОГО УРАВНЕНИЯ $d^2y/dx^2 = f(x, y, y')$ С НАЧАЛЬНЫМИ УСЛОВИЯМИ КОШИ $y(x_0) = y_0$; $y'(x_0) = y'_0$. ИСПРАВЛЕННЫЙ МЕТОД ЭЙЛЕРА. ФОРМУЛЫ (3.36)

	0	1	2	3	4	5
0	В/0 48	ПП 68	Р5 51	F4 42	↔ 76	+ 96
1	Р4 41	+ 96	Р5 51	↑ 06	F6 62	× 26
2	2 24	÷ 36	↑ 06	F3 32	+ 96	Р3 31
3	F6 62	↑ 06	F2 22	+ 96	Р2 21	ПП 68
4	Р5 51	F5 52	+ 96	2 24	÷ 36	Р4 41
5	С/П 78					
6						
7			$f(x,y,y')$	25 шагов		
8						
9			↑ 06	F6 62	× 26	↑ 06

	x_0+h	y_1	y_1'			$f(x, y, y')$	
Регистры памяти	2	3	4	5	6	7	8
	x_0	y_0	y_0'		h		

Порядок работы

1. *Программирование* $f(x, y, y')$ — правой части (3.31). Начало подпрограммы БП P5 . Переменные x, y и y' берутся соответственно из регистров памяти <2>, <3> и <4>. Для программирования $f(x, y, y')$ свободны регистры <7>, <8>. Результат вычисления $f(x, y, y')$ оставляется в регистре <X>. При короткой подпрограмме $f(x, y, y')$ добавляется в ее конце команда безусловного перехода БП $F9$.
2. *Ввод шага* h в <6>.
3. *Ввод* x_0 в <2>.
4. *Ввод* y_0 в <3>.
5. *Ввод* y_0' в <4>.
6. *Пуск* В/О С/П .
7. *Результаты*: После останова в регистре <2> оказывается величина $x_1 = x_0 + h$; в регистре <3> величина $y_1 = y(x_0 + h)$; в регистре <4> величина $y_1' = y'(x_0 + h)$.
8. *Повторный пуск* В/О С/П . При этом все величины, указанные в п. 7, смещаются на шаг и т. д.

Примеры: 1. Уравнение $y'' = y$ при начальных условиях $y(0) = 1$; $y'(0) = -1$.

Точное решение: $y_T = e^{-x}$.

Результаты ($h = 0{,}05$):

x	0,5	1,0	1,5	2,0	2,5	3,5	4,5
y	0,60673	0,36829	0,22383	0,13650	0,08400	0,0354	0,0252
y_T	0,60654	0,36788	0,22313	0,13534	0,08209	0,0302	0,0111

Устойчивость решения нарушается, начиная примерно с $x = 4{,}5$, когда прослеживается появление растущей экспоненты (напомним, что общим решением рассматриваемого уравнения с указанными граничными условиями является $y = Ce^x + e^{-x}$, где C — малая константа порядка погрешностей округления). Правильный ход решения до относительно большой величины x указывает на сравнительно хорошую устойчивость схемы (3.30).

2. Уравнение

$$y'' - 2y' + y = e^{-x} \sin x + 4e^x \tag{3.41}$$

с граничными условиями $y(0) = 0{,}16$; $y'(0) = 0{,}96$.

Точное решение

$$y_T = x e^x + 2x^2 e^x + (3 \sin x + 4 \cos x) e^{-x}/25.$$

Подпрограмма правой части уравнения (3.41), разрешенного относительно y'', занимает 23 шага:

F2 P÷ P8 F, ↑ F2 P+ × P7 F8 4 ×

↑ F7 + ↑ F3 − ↑ F4 $\overset{\leftarrow}{\underset{\rightarrow}{xy}}$ + +

Результаты ($h = 0,05$):

x	0,2	0,3	0,4	0,5	0,7	1,0	1,5	2,0
y	0,48814	0,78426	1,1991	1,7612	3,4672	8,1880	26,779	73,547
y_T	0,48990	0,78744	1,2042	1,7688	3,4823	8,2238	26,919	73,896

Общее время счета (40 шагов) около 15 мин.

3.6. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ МНОГОЧЛЕНОВ ДЕСЯТОЙ СТЕПЕНИ

Далее приведена программа расчета значений многочленов

$$F(x) = a_0 + a_1x + \dots + a_{10}x^{10}. \quad (3.42)$$

Вычисления осуществляются по известной схеме:

$$F(x) = (\dots((a_{10}x + a_9)x + a_8)x + \dots + a_1)x + a_0, \quad (3.43)$$

которая, в принципе, не нуждается в регистрах памяти для хранения промежуточных результатов. Программа пригодна и для расчета многочленов, степень которых меньше десяти. Для этого следует ввести нулевые значения соответствующих коэффициентов.

ПРОГРАММА 48

МНОГОЧЛЕНЫ ДЕСЯТОЙ СТЕПЕНИ. ФОРМУЛА (3.43)

	0	1	2	3	4	5
0		↑ 06	F2 22	× 26	↑ 06	F7 72
1	+ 96	↑ 06	F2 22	× 26	↑ 06	F6 62
2	+ 96	↑ 06	F2 22	× 26	↑ 06	F5 52
3	+ 96	↑ 06	F2 22	× 26	↑ 06	F4 42
4	+ 96	↑ 06	F2 22	× 26	P8 81	P× 23
5	FВП 65	FВП 65	FВП 65	P/−/ 53	P3 31	↑ 06
6	P× 23	FВП 65	FВП 65	FВП 65	− 86	PБП 59
7	P8 81	F8 82	↑ 06	F2 22	÷ 36	P3 31
8	С/П 78	F3 32	↑ 06	F8 82	+ 96	↑ 06
9	F2 22	× 26	P8 81	F3 32	БП 58	P/−/ 53

	$F(x)$						
Регистры памяти	2	3	4	5	6	7	8
	x		a_6	a_7	a_8	a_9	

Порядок работы

1. *Ввод переменной* x в <2>.

2. *Ввод коэффициентов* $a_0, a_1, \ldots, a_{10}$ в память калькулятора. Для хранения коэффициентов наряду с обычными регистрами памяти используются также шесть регистров стека, куда вводятся значения $a_0, a_1, \ldots, a_5$. Полная система команд, обеспечивающая ввод значений всех коэффициентов, имеет следующий вид:

$$\widehat{a_0}\ \ P,\ \ \widehat{a_1}\ \ P,\ \ \widehat{a_2}\ \ P,\ \ \widehat{a_3}\ \ P,\ \ \widehat{a_4}\ \ P,\ \ \widehat{a_5}\ \ P,\ \ \widehat{a_6}\ \ P4\ \ \widehat{a_7}\ \ P5$$

$$\widehat{a_8}\ \ P6\ \ \widehat{a_9}\ \ P7\ \ \widehat{a_{10}}.$$

Здесь $\widehat{a_0}, \widehat{a_1}, \ldots, \widehat{a_{10}}$ означает операции набора на клавиатуре численных значений соответствующих констант. $P,$ — наименование команды ввода числа в стек по часовой стрелке (см. Введение, § 1). Последний коэффициент a_{10} вводится в регистр <X>, т. е. после набора на клавиатуре оставляется в регистре <X>.

3. *Пуск* В/О С/П .

4. *Результат*: Численное значение многочлена выдается на индикатор и засылается в регистр <3>.

5. *При повторении вычислений* с другими x требуется занести новое значение x в регистр <2>, набрать на клавиатуре снова значение коэффициента a_{10} и затем осуществить пуск В/О С/П . Содержание остальных регистров памяти и стека не должно изменяться.

Время счета примерно 30 с.

Примечание:

— если члены с некоторыми степенями x^i отсутствуют, следует вводить нулевые значения соответствующих коэффициентов a_i.

Пример. Вычислить значения многочлена $F(x) = 1 + 2x + 3x^2 + \ldots + 11x^{10} = (11x^{12} - 12x^{11} + 1)/(x - 1)^2$.

$F(3) = 930022$. $F(4) = 14913081$. Результаты точные.

3.7. ИНТЕРПОЛЯЦИЯ

Ниже приведены две программы вычисления значений функций по интерполяционным формулам. Значения интерполяционных многочленов рассчитываются по известным значениям функций в заданных точках (узлах интерполяции).

В программе 49 реализована квадратичная интерполяция при значениях аргумента, которые могут быть и не равноотстоящими. Вычисления выполняются по формуле

$$F(x) = F_3 + (x - x_3)\,[\alpha_2 + (\alpha_1 - \alpha_2)\,(x - x_2)/\,(x_1 - x_2)], \qquad (3.44)$$

где $\alpha_1 = (F_3 - F_1)/\,(x_3 - x_1)$; $\alpha_2 = (F_3 - F_2)/\,(x_3 - x_2)$. Величины F_1, F_2, F_3 — известные значения функции в точках x_1, x_2, x_3:

$$F(x_1) = F_1;\ F(x_2) = F_2;\ F(x_3) = F_3.$$

В программе 50 реализовано интерполирование по формуле Лагранжа при пяти точках (равноотстоящие узлы). Формула имеет вид (см., например, [4]).

$$F(x) \equiv F(x_0 + \Delta x) = \frac{1}{24}(B_{-2}F_{-2} - 4B_{-1}F_{-1} + 6B_0F_0 - 4B_1F_1 + B_2F_2), \tag{3.45}$$

где

$$B_k = p(p^2-1)(p^2-4)/(p-k), \quad k = -2;\ -1;\ 0;\ 1;\ 2, \tag{3.46}$$

$p = \Delta x/h$, h — шаг; F_k — значения функции $F(x)$ в узлах x_k, т. е. $F_k = F(x_k) = F(x_0 + kh)$, $k = -2;\ -1;\ 0;\ 1;\ 2$.

ПРОГРАММА 49

КВАДРАТИЧНАЯ ИНТЕРПОЛЯЦИЯ. ФОРМУЛА (3.44)

	0	1	2	3	4	5
0	С/П 78	F5 52	/-/ 56	↑ 06	F2 22	+ 96
1	Р, 43	Р, 43	F4 42	+ 96	Р, 43	F3 32
2	+ 96	Р, 43	F4 42	/-/ 56	↑ 06	F3 32
3	+ 96	Р, 43	F2 22	+ 96	Р, 43	F7 72
4	↑ 06	F8 82	− 86	↑ 06	Р, 43	Р, 43
5	Р, 43	÷ 36	Р2 21	F6 62	↑ 06	F8 82
6	− 86	↑ 06	Р, 43	÷ 36	↑ 06	F2 22
7	− 86	↑ 06	Р, 43	÷ 36	↑ 06	Р, 43
8	× 26	↑ 06	F2 22	+ 96	↑ 06	Р, 43
9	Р, 43	× 26	↑ 06	F8 82	+ 96	Р2 21

	$F(x)$						
Регистры памяти	2	3	4	5	6	7	8
	x	x_1	x_2	x_3	F_1	F_2	F_3

Порядок работы

1. *Ввод* x_1—x_3 соответственно в регистры <3> ... <5>.
2. *Ввод* F_1—F_3 соответственно в <6> ... <8>.
3. *Ввод* x в <2>.

4. *Пуск* В/О С/П .

5. *Результат*: Значение функции $F(x)$ на индикаторе и в регистре <2>.

6. *Для вычисления* $F(x)$ при новом значении x и прежних узлах интерполяции достаточно ввести новое x в <2> и затем пуск В/О С/П .

Время счета примерно 10 с.

Пример. Вычислить значения функций Бесселя интерполированием.

Зададимся значениями $J_0(x)$ в узлах $x=1{,}5$; $1{,}8$; $2{,}3$, воспользовавшись таблицей $J_0(x)$ из [4]: $J_0(1{,}5) = 0{,}511828$; $J_0(1{,}8) = 0{,}339986$; $J_0(2{,}3) = 0{,}055540$.

Результаты интерполяции:

$J_0(1{,}7) = 0{,}3971687$. Табличное значение 0,397985;

$J_0(1{,}9) = 0{,}2829010$. Табличное значение 0,281819.

ПРОГРАММА 50

ПЯТИТОЧЕЧНАЯ ИНТЕРПОЛЯЦИЯ. ФОРМУЛА ЛАГРАНЖА (3.45), (3.46)

	0	1	2	3	4	5
0		0 04	Р5 51	F2 22	↑ 06	F7 72
1	÷ 36	Р2 21	1 14	− 86	1 14	− 86
2	Р8 81	× 26	Р3 31	F2 22	1 14	+ 96
3	1 14	+ 96	Р4 41	× 26	↑ 06	F2 22
4	× 26	↑ 06	F3 32	× 26	Р3 31	F3 32
5	↑ 06	F4 42	÷ 36	↑ 06	F6 62	Р, 43
6	Р6 61	× 26	↑ 06	F5 52	+ 96	Р5 51
7	F4 42	1 14	− 86	Р4 41	↑ 06	F8 82
8	− 86	РПП 69	F, 45	F6 62	Р, 43	Р, 43
9	F5 52	2 24	4 44	÷ 36	Р2 21	С/П 78

	2	3	4	5	6	7	8
	$F(x)$						
Регистры памяти	2	3	4	5	6	7	8
	$x-x_0$					h	

Порядок работы

1. *Ввод* $(x - x_0)$ в <2>, где x — аргумент и x_0 — центральный узел интерполяции.

2. *Ввод шага* h в <7>.

3. *Ввод в стек* значений функций в пяти узлах интерполяции:

$F(x_0 - 2h)$; $-4F(x_0 - h)$; $6F(x_0)$; $-4F(x_0 + h)$; $F(x_0 + 2h)$.

Ввод осуществляется последовательным нажатием на следующие клавиши: $1F_{-2}$ Р, $-4F_{-1}$ Р, $6F_0$ Р, $-4F_1$ Р, $1F2$ Р, Р. Здесь MF_h ($M = 1; -4; 6; -4; 1$; $k = -2; -1; 0; 1; 2$) — означает набор на клавиатуре соответствующих численных значений.

4. *Пуск* В/О С/П .

5. *Результат*: Значение $F(x)$ в регистре <2> и на индикаторе. *Время счета* примерно 30 с.

6. *Вычисление $F(x)$ при других x* и тех же узлах интерполяции производится вводом в регистр <2> нового значения $(x - x_0)$ и последующим пуском В/О С/П . Результат фиксируется на индикаторе и в регистре <2>. Между последовательными пусками, соответствующими разным x, *недопустимо* на калькуляторе производить вычисления с использованием встроенных функций $\sin x$, $\cos x$, e^{ix}, $\ln x$, e^x, x^y. Если же такие операции необходимы, то после них следует повторить п. 3, так как указанные операции могут стирать числа, хранящиеся в стеке.

Пример. Вычислить $\sin x$ интерполированием.

Зададимся узлами интерполяции — 30, 0, 30, 60, 90°, в которых $\sin x$ принимает известные значения: $-0{,}5$; 0; 0,5; $\sqrt{3}/2$; 1. Вводим в стек величины (см. п. 3): $-0{,}5$; 0; 3; $-2\sqrt{3}$; 1. Вводим в <7> шаг $h = 30°$.

Результаты вычислений:

$\sin(15°) = 0{,}2592147$. Табличное значение 0,258819;

$\sin(45°) = 0{,}7067306$. Табличное значение 0,7071067;

$\sin(80°) = 0{,}9857018$. Табличное значение 0,984807.

Несмотря на довольно грубую сетку (пять значений функции охватывают практически весь диапазон изменения $\sin x$), погрешность не более 0,1%.

3.8. ПОИСК МИНИМУМА ФУНКЦИИ

Здесь приведена программа вычисления положения и значения минимума функции $y = f(x)$ в заданном интервале (x_0, L) независимого переменного. Расчет ведется методом «золотого сечения» (см., например, [9]).

Метод состоит в следующем. Заданный интервал делится в «золотом сечении», т. е. находятся точки x_2 и x_3, для которых $x_2 - x_0 = L - x_3 = (3 - \sqrt{5})(L - x_0)/2 \approx 0{,}38(L - x_0)$. Далее сравниваются значения $f(x_2)$ и $f(x_3)$. Допустим, что $f(x_2) > f(x_3)$. Тогда интервал поиска сужается до (x_2, L) и величина x_0 заменяется на x_2, после чего процесс повторяется. Аналогичная ситуация имеет место при $f(x_2) < f(x_3)$ с заменой L на x_3.

Программа пригодна для нахождения действительных корней уравнения $\varphi(x) = 0$, являющихся минимумами функции $f(x) = (\varphi(x))^2$.

ПРОГРАММА 51

ПОИСК МИНИМУМА ФУНКЦИИ $y=f(x)$. МЕТОД «ЗОЛОТОГО СЕЧЕНИЯ» [9]

	0	1	2	3	4	5
0	В/О 48	F2 22	P5 51	F8 82	1 14	— 86
1	P8 81	FВП 65	F4 42	↑ 06	F5 52	— 86
2	8 84	6 64	P— 83	× 26	↑ 06	F4 42
3	+ 96	P3 31	F5 52	— 86	/—/ 56	P2 21
4	ПП 68	PВП 63	P7 71	F3 32	ПП 68	PВП 63
5	↑ 06	F7 72	— 86	PВ/О 49	P0 01	F3 32
6	P4 41	БП 58	P↑ 03			
7						
8			$f(x)$	21 шаг		
9						

	x_2	x_3			$f(x)$	y_{min}	
Регистры памяти	2	3	4	5	6	7	8
	x_0		L				N

Порядок работы

1. *Программирование* $f(x)$. Подпрограмма начинается с ячейки 63 (БП РВП). Аргумент функции $f(x)$ к началу подпрограммы содержится в регистре <X>. Для программирования можно использовать регистр <6>. Кроме того, для хранения промежуточных результатов в пределах рабочего цикла подпрограммы можно использовать стек (ограничения при использовании стека см. в комментариях и к программе 35). Результат вычисления $f(x)$ оставляется в регистре <X>. При короткой подпрограмме в ее конце добавляется команда безусловного перехода БП + .
2. *Ввод* x_0 в <2>.
3. *Ввод* L в <4>.
4. *Ввод числа итераций* N в <8>.
5. *Пуск* В/О С/П . При останове — индикация переполнения.
6. *Результаты*: Положение минимума определяется как полусум-

ма величин x_2 и x_3, содержащихся соответственно в регистрах <2> и <3>. Разность $(x_3 - x_2)$ определяет погрешность найденного значения x_{min}. Значение минимума y_{min} содержится в регистре <7>.

7. Если точность вычисления x_{min} недостаточна и необходимо выполнить дополнительное число ΔN итераций, то в регистр <8> вводится число $(\Delta N - 1)$ и затем осуществляется пуск F4 С/П .

Пример. Найти минимум функции $y = 3 - 2x + x^2$ в диапазоне (0; 4). Вводим подпрограмму $y(x)$: 2 /—/ × $\overleftarrow{\underrightarrow{xy}}$ F/—/ + 3 + БП + . Как видно, длина подпрограммы 8 шагов, не считая команды безусловного перехода БП + , которая не является обязательной.

Результаты вычислений при разном числе итераций:

$x_2 = 0{,}9458969$; $x_3 = 1{,}519304$; $x_{min} = 1{,}232600$; $y_{min} = 2{,}002927$ при $N = 2$;

$x_2 = 0{,}8043786$; $x_3 = 0{,}9386058$; $x_{min} = 0{,}8714922$; $y_{min} = 2{,}038267$ при $N = 5$;

$x_2 = 0{,}9924724$; $x_3 = 1{,}004406$; $x_{min} = 0{,}9984396$; $y_{min} = 2{,}000056$ при $N = 10$;

$x_2 = 0{,}9988313$; $x_3 = 0{,}9998925$; $x_{min} = 0{,}9993619$; $y_{min} = 2{,}000001$ при $N = 15$.

Точное значение $x_{min} = 1$; $y_{min} = 2$.

Время счета $t \approx (N/4)$ мин.

3.9. ПЕРЕВОД ЧИСЕЛ ИЗ ДЕСЯТИЧНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В СИСТЕМУ С ОСНОВАНИЕМ, МЕНЬШИМ ДЕСЯТИ

Рассмотрим перевод числа A_{10}, записанного в десятичной системе счисления, в число A_k в системе с основанием $k < 10$. Известные алгоритмы последовательного деления и умножения (см., например, [14]) при всей их простоте оказываются неудобными для непосредственной реализации в микрокалькуляторе «Электроника Б3-21». Ниже используется следующая процедура.

Находится максимальный показатель степени n, для которого справедливо неравенство $k^n \leqslant A_{10}$.

Производится вычитание k^n из A_{10} p_n раз, где $p_n k^n < A_{10} < (p_n + 1)k^n$. Число $p_n < k$ является цифрой старшего разряда A_k.

Выполняются те же действия, но теперь роль числа A_{10} играет разность $A_{10} - p_n k^n$, а вместо k^n берется k^{n-1}. В результате получается следующая цифра A_k. Указанный процесс повторяется до исчерпания целой части A_{10} и дает целую часть A_k.

Дробная часть числа преобразуется точно так же, но теперь в вычислениях роль основания системы играет k^{-8} (степень 8 соответствует числу значащих цифр у микрокалькулятора). В результате находится дробная часть числа A_k в нормализованном виде с 7 значащими цифрами.

Максимальное число, которое может быть преобразовано в k-систему, равно k^8, не считая дробной части. Исключение составляет двоичная система (см. далее).

ПРОГРАММА 52

ПЕРЕВОД ЧИСЛА A_{10} ИЗ ДЕСЯТИЧНОЙ СИСТЕМЫ В СИСТЕМУ С ОСНОВАНИЕМ k, МЕНЬШИМ ДЕСЯТИ ($A_{max}=k^8$)

	0	1	2	3	4	5
0		P, 43	1 14	P8 81	F7 72	F/—/ 55
1	F/—/ 55	F/—/ 55	P3 31	1 14	0 04	P6 61
2	F/—/ 55	F/—/ 55	F/—/ 55	P4 41	F, 45	P, 43
3	0 04	P5 51	F2 22	↑ 06	F8 82	— 86
4	РПП 69	5 54	P/—/ 53	P8 81	P/—/ 53	P2 21
5	F5 52	С/П 78	БП 58	0 04	F2 22	↑ 06
6	F3 32	— 86	РПП 69	8 84	F4 42	↑ 06
7	F6 62	÷ 36	P4 41	F3 32	↑ 06	F7 72
8	÷ 36	P3 31	БП 58	5 54	P2 21	F5 52
9	↑ 06	F4 42	+ 96	P5 51	БП 58	F3 32

Регистры памяти	2	3	4	5	6	7	8
	$A_{10(цел)}$					k	1

Порядок работы

1. *Ввод* основания k новой системы в <7>.
2. *Ввод* целой части числа $A_{10\,цел}$ в регистр <2>.
3. *Набор* дробной части числа A_{10} на клавиатуре. Дробную часть можно набирать и в нормализованном виде.
4. *Пуск* В/О С/П . После останова калькулятора на индикаторе и в регистре <5> оказывается целая часть числа A_k. Если исходное число было равно k^8, т. е. значению допустимого максимума, то на индикаторе будет число $1 \cdot 10^8$.
5. *Повторный пуск* С/П . После останова на индикаторе и в регистре <5> окажется дробная часть A_k в нормализованном виде: двузначное отрицательное число в регистрах 10—12 обозначает порядок числа в системе k, т. е. отрицательную степень числа k.

Максимальное время счета при преобразовании целой части числа примерно 4 мин (преобразование числа $9^8 - 1 = 43046720$ в девятиричную систему). Таким же является максимальное время счета при преобразовании дробной части (преобразование в девятиричную систему числа 0,99999). При уменьшении k максимальное время счета также уменьшается примерно пропорционально k.

Если число, которое требуется преобразовать в двоичную систему, превышает $2^8 = 256$ или требуется большее число значащих цифр в дробной части двоичного числа, то A_{10} сначала можно преобразовать в восьмеричное число. Затем каждую цифру последнего следует представить в двоичном коде, что легко выполняется без применения вычислительных средств.

Пример. Представить в двоичном коде число $A_{10} = 247{,}119$. Вводим 2 в <7>, 247 в <2> и набираем 0,119. После первого пуска получаем число 11110111 — число 247 в двоичном коде. После второго пуска $1{,}111001 \cdot 2^{-4}$ двоичное представление 0,119. Таким образом, 247,119 = 11110111, 0001111001 ...

Проделаем ту же операцию, используя восьмеричный код. Заменяем основание 2 в регистре <7> на 8. Снова вводим 247 в <2> и набираем 0,119. После двух пусков получаем число 247,119 в восьмеричном коде: 367,07473310. Представим каждую цифру полученного числа в двоичном коде. Окончательно двоичное представление заданного числа имеет вид: 247,119 = 11110111, 000111100111011011001...

3.10. ДЕЙСТВИЯ НАД КОМПЛЕКСНЫМИ ЧИСЛАМИ

Здесь приведены две программы для выполнения некоторых операций над комплексными числами. В программе 53 вычисляются произведение, частное и модули комплексных чисел $A = a_1 + ia_2$ и $B = b_1 + ib_2$

$$A \cdot B = c_1 + ic_2;\ A/B = d_1 + id_2;\ r_A = \sqrt{a_1^2 + a_2^2};\ r_B = \sqrt{b_1^2 + b_2^2}.$$

Произведение и частное находятся по обычным формулам:

$$c_1 = a_1b_1 - a_2b_2;\ c_2 = a_1b_2 + a_2b_1;$$

$$d_1 = (a_1b_1 + a_2b_2)/(b_1^2 + b_2^2);\ d_2 = (a_2b_1 - a_1b_2)/(b_1^2 + b_2^2).$$

В программе 54 производится преобразование комплексного числа к тригонометрической форме и вычисляется степенна́я функция A_n, где n — вещественное число. В первом случае

$$A = a + ib = re^{i\varphi},$$

где $r = \sqrt{a^2 + b^2}$ — модуль A. В качестве аргумента φ выбирается функция

$$\varphi = \arg(A) = \begin{cases} \arccos(a/r), & b \geqslant 0; \\ 2\pi - \arccos(a/r), & b < 0. \end{cases} \tag{3.47}$$

Таким образом, комплексная плоскость предполагается разрезанной по вещественной полуоси $(x > 0)$ и $0 \leqslant \varphi < 2\pi$. A_n вычисляется по формуле

$$A_n = a_n + ib_n = r^n \cos(n\varphi) + ir^n \sin(n\varphi). \tag{3.48}$$

Если n — дробное число, то в качестве A^n выбирается согласно (3.47) и (3.48), та ветвь, для которой $1^n = 1$. Для получения других ветвей A^n к φ из (3.47) следует добавить $2\pi(m - 1)$, где m — номер ветви, и затем вновь использовать (3.48).

ПРОГРАММА 53

ПРОИЗВЕДЕНИЕ, ЧАСТНОЕ, МОДУЛИ КОМПЛЕКСНЫХ ЧИСЕЛ

	0	1	2	3	4	5
0	$\boxed{\text{С/П} \ 78}$	F2 22	F/-/ 55	↑ 06	F3 32	F/-/ 55
1	+ 96	FВП 65	P, 43	F5 52	↑ 06	F3 32
2	× 26	P6 61	F2 22	× 26	P7 71	F5 52
3	F/-/ 55	↑ 06	F4 42	F/-/ 55	+ 96	F, 45
4	P8 81	F4 42	↑ 06	F3 32	× 26	P4 41
5	F2 22	× 26	↑ 06	F6 62	− 86	P2 21
6	F6 62	+ 96	P6 61	F4 42	↑ 06	F7 72
7	+ 96	P3 31	F7 72	− 86	↑ 06	F8 82
8	$\overleftarrow{\underrightarrow{xy}}$ 16	× 26	P5 51	F6 62	× 26	P4 41
9	$\overleftarrow{\underrightarrow{xy}}$ 16	F, 45	FВП 65	P8 81	P/-/ 53	P7 71

	c_1	c_3	d_1	d_2		r_A	r_B
Регистры памяти	2	3	4	5	6	7	8
	a_1	a_2	b_1	b_2			

Порядок работы

1. *Ввод* действительной и мнимой частей комплексного числа $A = a_1 + ia_2$ соответственно в регистры $<2>$ и $<3>$.
2. *Ввод* действительной и мнимой частей $B = b_1 + ib_2$ в регистры $<4>$ и $<5>$.
3. *Пуск* В/О С/П.
4. *Результаты:* В регистрах $<2>$ и $<3>$ соответственно действительная и мнимая части произведения $A \cdot B = c_1 + ic_2$; в регистрах $<4>$ и $<5>$ действительная и мнимая части частного $A/B = d_1 + id_2$ в регистрах $<7>$ и $<8>$ соответственно модули r_A и r_B.
5. *Погрешности* не превышают одной единицы младшего разряда.
6. *Время счета* примерно 15 с.

Пример. Вычислить произведение, частное и модули комплексных чисел $A = 5 + 2i$; $B = 3 + 4i$.

$$A \cdot B = 7 + 26i; \ A/B = 0{,}92 - 0{,}56i;$$

$$r_A = 5{,}385164; \ r_B = 5.$$

ПРОГРАММА 54

ТРИГОНОМЕТРИЧЕСКАЯ ФОРМА И СТЕПЕНЬ КОМПЛЕКСНОГО ЧИСЛА

	0	1	2	3	4	5
0	$\boxed{\text{С/П} \quad 78}$	F2 22	F/–/ 55	↑ 06	F3 32	F/–/ 55
1	+ 96	FВП 65	P4 41	F2 22	9 94	P7 71
2	F4 42	÷ 36	P8 81	1 14	P5 51	F5 52
3	P– 83	↑ 06	F8 82	– 86	↑ 06	F5 52
4	P+ 93	÷ 36	↑ 06	F5 52	+ 96	P5 51
5	F7 72	1 14	– 86	P7 71	РПП 69	F× 25
6	F3 32	РПП 69	PCx 73	P× 23	↑ 06	F5 52
7	– 86	+ 96	P5 51	F6 62	↑ 06	F5 52
8	× 26	P7 71	F4 42	x^y 38	↑ 06	F7 72
9	P– 83	× 26	P8 81	F7 72	P+ 93	× 26

			r	φ			a_n
Регистры памяти	2	3	4	5	6	7	8
	a	b			n		

Порядок работы

а) $|b/a| \geqslant 0{,}05$

1. *Ввод* в регистры <2> и <3> соответственно действительной и мнимой частей числа $A = a + b\mathrm{i}$.
2. *Ввод* в регистр <6> целого или дробного показателя степени n.
3. *Пуск* В/О С/П .
4. *Результаты*: В регистрах <4> и <5> соответственно модуль r и аргумент φ числа A. В регистре <8> и на индикаторе соответственно действительная a_n и мнимая b_n части числа $A^n = (a + b\mathrm{i})^n$.

Время счета примерно 1,5 мин.

б) $|b/a| < 0{,}05$.

Предыдущая процедура начинает приводить к сравнительно большим погрешностям φ, a_n, b_n, возрастающим с уменьшением b. Так, при $b=0$ относительная погрешность φ достигает $\approx 2\cdot 10^{-4}$. Точность повышается, если вычислять φ по формулам, которые следуют из разложения для $\varphi_0 = \operatorname{arctg}(b/a)$ при малых $|b/a|$:

$$\left.\begin{aligned} &\varphi=\varphi_0 \approx b\cdot(1-b^2/(3a^2))/a, \quad a>0;\; b\geqslant 0;\\ &\varphi=\varphi_0+2\pi, \quad a>0;\quad b<0;\\ &\varphi=\varphi_0+\pi, \quad a<0. \end{aligned}\right\} \tag{3.49}$$

Чтобы внести поправки в φ, a_n и b_n, связанные с использованием (3.49), после п. 1—3 нужно выполнить следующие операции.

5. *Вычислить* нужную из величин (3.49) и ввести в регистр <5>. Эти действия проще всего выполнить нажатием следующих клавиш:

$a>0;\ b\geqslant 0$: $F3$ ↑ $F2$ ÷ P7 F/—/ 3 ÷ /—/ 1 + ↑ $F7$ × P5

$a>0;\ b<0$: $F3$ ↑ $F2$ ÷ P7 F/—/ 3 ÷ /—/ 1 + ↑ $F7$ × ↑ P× $\overleftarrow{\underrightarrow{xy}}$ + + P5 .

$a<0$: $F3$ ↑ $F2$ ÷ P7 F/—/ 3 ÷ /—/ 1 + ↑ $F7$ × ↑ P× + P5 .

6. *Пуск* БП PC_x С/П .

Время счета примерно 10 с.

7. *Результаты*: В регистрах <4> и <5> соответственно модуль r и аргумент φ числа А. В регистре <8> действительная часть a_n числа A^n; на индикаторе мнимая часть b_n.

8. *Вычисление следующих ветвей* A^n. После выполнения п. 3 при $|b/a|\geqslant 0{,}05$ или после п. 5 для $|b/a|<0{,}05$ к содержимому регистра <5> добавляется 2π и далее осуществляется пуск БП PC_x СП . Указанные действия производятся нажатием следующих клавиш: P × ↑ $F5$ + + P5 БП PC_x С/П . Они могут повторяться до исчерпания всех ветвей, если показатель степени n является рациональной дробью.

Время счета примерно 10 с.

Погрешности. Погрешность модуля r не более единицы в последней цифре младшего разряда. Относительная погрешность φ менее $1\cdot 10^{-5}$. Абсолютные погрешности a_n и b_n меньше $na_{\max}\cdot 10^{-5}$, где n — показатель степени и $a_{\max}$—наибольшее из чисел a_n или b_n.

Пример. Преобразовать к тригонометрической форме число $A = 2 + 11i$. Вычислить кубический корень A.

Результаты:

$r = 11{,}18033$; $\varphi = 1{,}390942$;

$(2+11i)_1^{1/3} = 1{,}999998 + 0{,}9999992i$;

$(2+11i)_2^{1/3} = -1{,}866024 + 1{,}2320501i$;

$(2+11i)_3^{1/3} = -0{,}1339752 - 2{,}2320491i$.

3.11. ЭЛЕМЕНТАРНЫЕ ФУНКЦИИ КОМПЛЕКСНОГО АРГУМЕНТА

Здесь даются программы для табулирования некоторых элемента ных функций комплексного аргумента. Программа 55 позволяет вы числять значения следующих функций аргумента $z = x + \mathrm{i}y$:

$$\left.\begin{aligned} \sin z &= s_1 + \mathrm{i}s_2 = \sin x \operatorname{ch} y + \mathrm{i} \cos x \operatorname{sh} y; \\ \cos z &= c_1 + \mathrm{i}c_2 = \cos x \operatorname{ch} y - \mathrm{i} \sin x \operatorname{sh} y; \\ \operatorname{tg} z &= t_1 + \mathrm{i}t_2 = \sin z / \cos z. \end{aligned}\right\} \quad (3.5$$

$$\left.\begin{aligned} \operatorname{sh} z &= s_{h1} + \mathrm{i}s_{h1} = \operatorname{sh} x \cos y + \mathrm{i} \operatorname{ch} x \sin y; \\ \operatorname{ch} z &= c_{h1} + \mathrm{i}c_{h2} = \operatorname{ch} x \cos y - \mathrm{i} \operatorname{sh} x \sin y; \\ \operatorname{th} z &= t_{h1} + \mathrm{i}t_{h2} = \operatorname{sh} z / \operatorname{ch} z. \end{aligned}\right\} \quad (3.5$$

В программах 56, 57 находятся значения комплексных многочлен

$$F(z) = c_n z^n + c_{n-1} z^{n-1} + \ldots + c_1 z + c_0; \qquad (3.5$$

$$z = x + \mathrm{i}y; \quad c_m = a_m + \mathrm{i}b_m \; (m = 0, 1, 2, \ldots, n).$$

Вычисления ведутся по известной схеме

$$F(z) = (\ldots (c_n z + c_{n-1}) z + c_{n-2}) z + \ldots) z + c_0 = F_1 + \mathrm{i}F_2.$$

Программа 56 рассчитана на табулирование многочленов до 4-й сте ни включительно. В ней обеспечивается сохранение коэффициент c_m (кроме c_n) в памяти калькулятора после каждого пуска. Програ ма 57 позволяет вычислять многочлены произвольной степени. Одна при каждом значении аргумента z должен заново выполняться пос довательный ввод всех коэффициентов c_m ($m = 0, 1, \ldots,$).

В отличие от предыдущих разделов книги здесь при записи програ табличная форма не используется. Последовательность команд в п граммах 55 — 57 записывается *построчно, начиная с ячейки* (ср. рис. 5), *так что переход в режим программирования осуществ ется нажатием клавиш* В/О Р $\overleftarrow{\text{ШГ}}$.

ПРОГРАММА 55

ТРИГОНОМЕТРИЧЕСКИЕ И ГИПЕРБОЛИЧЕСКИЕ ФУНКЦИИ КОМПЛЕКСНОГО АРГУМЕНТА. ФОРМУЛЫ (3.50), (3.51)

P÷	↑	F,	P6	—	2	÷	P7	↑	F6	+
P8	F2	P↑	P4	F8	×	P3	F7	×	/—/	P6
F4	↑	F7	×	P4	F8	×	P5	↑	F3	×
P7	F4	×	P8	F6	↑	F3	×	P/—/	F4	×
↑	F7	+	P7	F8	↑	P,	—	P8	F6	F/—
↑	F5	F/—/	+	С/П						

sh, ch, th:	$-y$	s_{h1}	s_{h2}	c_{h1}	c_{h2}	t_{h1}	t_{h2}
Регистры памяти	2	3	4	5	6	7	8
sin, cos, tg:	x	s_1	s_2	c_1	c_2	t_1	t_2

Порядок работы

а) Вычисление тригонометрических функций

1. *Ввод* x в <2> и набор на клавиатуре y.

2. *Пуск* В/О С/П. После останова, не меняя числа на индикаторе, выполнить следующие команды:

$$F, \quad \uparrow \quad F7 \quad \times \quad Р7 \quad F8 \quad \times \quad Р8$$

3. *Результат*: значения действительной и мнимой частей sin z соответственно в регистрах <3> и <4>, действительной и мнимой частей cos z — в регистрах <5> и <6>, действительной и мнимой частей tg z — в регистрах <7> и <8>.

Время счета примерно 30 с.

Пример: вычислить тригонометрические функции аргумента $z=2+3i$.
$\sin z=9{,}154496-4{,}168903i$;
$\cos z=-4{,}189622-9{,}109225i$;
$\mathrm{tg}\, z=-3{,}764016\cdot 10^{-3}+1{,}003238i$;

б) Вычисление гиперболических функций

1. *Ввод* $(-y)$ в <2> и набор x на клавиатуре.

2. *Пуск* В/О С/П. После останова произвести дополнительные операции, не меняя числа на индикаторе:

$$F, \quad \uparrow \quad F8 \quad \times \quad Р2 \quad F7 \quad /-/ \quad \times \quad Р8 \quad F2$$
$$Р7 \quad F3 \quad /-/ \quad Р2 \quad F4 \quad Р3 \quad F2 \quad Р4$$

3. *Результат*: значения действительной и мнимой частей sh z соответственно в регистрах <3> и <4>, действительной и мнимой частей ch z — соответственно в регистрах <5> и <6>, действительной и мнимой частей th z — соответственно в регистрах <7> и <8>.

Пример: вычислить гиперболические функции аргумента $z=2+3i$.
$\mathrm{sh}z=-3{,}590564+0{,}5309206i$;
$\mathrm{ch}z=-3{,}724545+0{,}5118221i$;
$\mathrm{th}z=0{,}9653857-9{,}884373\cdot 10^{-3}i$.

ПРОГРАММА 56

КОМПЛЕКСНЫЕ ПОЛИНОМЫ 4-й СТЕПЕНИ. ФОРМУЛА (3.52)

Р4	$\underrightarrow{\overleftarrow{xy}}$	Р5	ПП	Р2	ПП	Р2	ПП	Р2	ПП	Р2
С/П	F3	↑	F5	×	Р8	F2	↑	F4	×	↑
F8	—	↑	F6	+	↑	F2	$\underrightarrow{\overleftarrow{xy}}$	Р2	F5	×
Р8	F3	↑	F4	×	↑	F8	+	↑	F7	+
Р3	F6	Р/—/	Р6	F7	Р/—/	Р7	В/О			

	F_1	F_2					
Регистры памяти	2	3	4	5	6	7	8
	a_4	b_4			a_3	b_3	

Порядок работы

1. *Ввод* коэффициентов a_4, b_4, a_3, b_3 в регистры соответствен <2>, <3>, <6>, <7>.

2. *Ввод* коэффициентов a_2, b_2, a_1, b_1, a_0, b_0 в стек посредством сл дующей системы команд:

$\widehat{a_2}$ P/—/ $\widehat{b_2}$ P/—/ $\widehat{a_1}$ P/—/ $\widehat{b_1}$ P/—/ $\widehat{a_0}$ P/—/ $\widehat{b_0}$ P/—

Здесь символы $\widehat{a_m}$ и $\widehat{b_m}$ означают набор соответствующих чисел н клавиатуре.

3. *Ввод* мнимой части y аргумента z в регистр <Y> и набор н клавиатуре действительной части x.

4. *Пуск* В/О С/П.

Время счета 40 с.

5. *Результат*: действительная и мнимая части многочлена F соответственно в регистрах <2> и <3>.

При повторении вычислений с новыми значениями аргумента сл дует восстановить содержимое регистров <2> и <3>, т. е. ввес в них соответственно a_4 и b_4, а затем ввести новые значения x и y с ответственно в регистры <X> и <Y>.

Внимание: если в промежутке между вычислениями $F(z)$ при разли ных z калькулятор использовался для нахождения функций x^y, ln e^{ix}, $\sin x$, $\cos x$, e^x, то необходимо повторить п. 2, т. е. восстанови содержимое стека.

Пример: вычислить значение многочлена

$$F(z)=(7+8i)\,z^4+(5+6i)\,z^3+(3+4i)\,z^2+(1+2i)\,z-(1+2i) \text{ при } z=9+1$$

$$F(z)=-188777-308378i.$$

ПРОГРАММА 57

КОМПЛЕКСНЫЕ МНОГОЧЛЕНЫ ПРОИЗВОЛЬНОЙ СТЕПЕНИ. ФОРМУЛА (3.52)

P6	$\overleftarrow{\underrightarrow{xy}}$	P7	F8	1	—	P8	F3	↑	F5	×
P/—/	F2	↑	F4	×	↑	P,	—	↑	F6	+
↑	F2	$\overleftarrow{\underrightarrow{xy}}$	P2	F5	×	P/—/	F3	↑	F4	×
↑	P,	+	↑	F7	+	P3	F8	С/П		
	F_1		F_2							

Регистры памяти	2	3	4	5	6	7	8
	a_n	b_n	x	y			n

Порядок работы

1. *Ввод* коэффициента $c_n = a_n + ib_n$ при старшем члене многочлена в регистры <2>, <3> (a_n — в регистр <2>, b_n — в регистр <3>).

2. *Ввод* аргумента $z = x + iy$ (x — в регистр <4>, y — в регистр <5>).

3. *Ввод* степени многочлена n в <8>.

4. *Ввод* коэффициента $c_{n-1} = a_{n-1} + ib_{n-1}$ (b_{n-1} — в регистр <Y>, a_{n-1} — в регистр <X>, т. е. набор числа a_{n-1} на клавиатуре.

5. *Пуск* В/О С/П.

6. *Ввод* следующего коэффициента $c_{n-2} = a_{n-2} + ib_{n-2}$ (b_{n-2} — в регистр <Y>, a_{n-2} — в регистр <X>).

7. *Пуск* В/О С/П.

8. Далее процесс повторяется: вводится очередной коэффициент в регистры <X> и <Y> и — пуск В/О С/П. При останове после очередного пуска на индикаторе появляется число, равное порядковому номеру предыдущего введенного коэффициента. Это число также хранится в регистре <8>.

9. *Результат*: действительная и мнимая части многочлена $F(z)$ — соответственно в регистрах <2> и <3>.

Время счета на одном шаге примерно 10 с.

Пример. Рассчитать значение многочлена шестой степени от аргумента $z = 1 - 2i$

$$F(z) = (1+2i)z^6 + (1-3i)z^5 + (1+4i)z^4 + (1+5i)z^3 + (2-i)z^2 + (6-5i)z + 5i.$$

$$F(z) = 222 + 311i.$$

СПИСОК ЛИТЕРАТУРЫ

1. **Трохименко Я. К., Любич Ф. Д.** Инженерные расчеты на микрокалькуляторах. — Киев: Техника, 1980. — 370 с.
2. **Градштейн И. С., Рыжик И. М.** Таблицы интегралов, сумм, рядов и произведений. — М.: Физматгиз, 1963. — 1100 с.
3. **Форсайт Дж., Малкольм М., Коулер К.** Машинные методы математических вычислений: Пер. с англ./Пер. Х. Д. Икрамова. — М.: Мир, 1980. — 210 с.
4. **Справочник** по специальным функциям: Пер. с англ./ Под ред. М. Абрамовица, И. Стиган; Пер. под ред. В. А. Диткина. — М.: Наука, 1979. — 832 с.
5. **Корн Г., Корн Т.** Справочник по математике: Пер. с англ./ Под ред. И. Г. Арамановича. — М.: Наука, 1973. — 832 с.
6. **Янке Е. Я., Эмде Ф., Леш Ф.** Специальные функции: Пер. с нем./Под ред. Л. И. Седова. — М.: Наука, 1968. — 344 с.
7. **Таблицы** значений функций Бесселя от мнимого аргумента. — М. — Л.: Изд. АН СССР, 1950. — 404 с.
8. **Hastings C., Hayward J.** Approximations for digital computers. — Princeton: Princeton Univ. Press, 1955.
9. **Калиткин Н. Н.** Численные методы. — М.: Наука, 1978. — 512 с.
10. **Мак-Кракен Д., Дорн У.** Численные методы и программирование на Фортране: Пер. с англ./ Под ред. Б. М. Наймарка. — М.: Мир, 1977. — 584 с.
11. **Холл Дж., Уатт Дж.** Современные численные методы решения обыкновенных дифференциальных уравнений: Пер. с англ./ Под ред. А. Д. Горбунова. — М.: Мир, 1979. — 312 с.
12. **Хемминг Р. В.** Численные методы: Пер. с англ. / Под ред. Р. С. Гутера. — М.: Наука, 1972. — 400 с.
13. **Камке Э.** Справочник по обыкновенным дифференциальным уравнениям: Пер. с нем./ Пер. В. А. Фомина. — М.: Наука, 1979. — 576 с.
14. **Гутер Р. С., Резниковский П. Т., Резник С. М.** Программирование и вычислительная математика, вып. 1. — М.: Наука, 1971. — 432 с.

Оглавление

35 к.

Special Functions
Programs for Microcalculator
(Elektronika B3-21)
Sh. E. Tsimring
1983

"РАДИО И СВЯЗЬ"