Министерство высшего и среднего специального образования РСФСР

ЛЕНИНГРАДСКИЙ ОРДЕНА ЛЕНИНА
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ имени М.И.КАЛИНИНА

ОСНОВЫ ПРОГРАММИРОВАНИЯ И ПРИМЕНЕНИЕ ЭВМ

Методические указания
к лабораторным работам I-3

Ленинград 1989

Основы программирования и применение ЭВМ: Метод.указ.к лабораторным работам I-3 /Сост.: Е.Д.Викторов, Г.Г.Гурьянова, В.Д.Дорот, В.И.Кубарев, А.Б.Смирнов, В.А.Шолуха.

Настоящие указания предназначены для студентов всех факультетов, изучающих курс "Основы программирования и применение ЭВМ" (здесь рассматривается часть курса - "Решение алгебраических трансцендентных уравнений. Приближение функций. Вычисление интегралов"). Предполагается, что студенты освоили первую часть курса, а именно основы программирования и у них имеются навыки в составлении схем и написании программ на языке Фортран.

Методические указания охватывают весь цикл лабораторных работ по методам вычислений, которые изучаются в разном объеме на всех факультетах. Каждая из работ содержит формулировку вычислительной проблемы и, как правило, несколько методов для ее решения. Для многих методов приведены схемы и программы их реализации на языке Фортран для ЭВМ "Искра-1256". В конце каждой работы даны задания.

Объем и характер приводимых в каждой работе сведений таков, что позволяет выполнить их без привлечения литературы. Однако ссылки на литературу, как основную, так и дополнительную, приведены в каждой работе.

Основы программирования и применение ЭВМ
Методические указания к лабораторным работам I-3

Научный редактор Л.Я.Болдырев
Редактор И.А.Лебедева
Технический редактор А.И.Колодяжная

Подписано к печати 19.09.88. Формат бумаги 60x90 1/16.
Бумага тип № 3. Печать офсетная. Усл.печ.л. 2,75. Уч.-изд.л. 2,5.
Тираж 2000. Заказ 389. Бесплатно.
Издание ЛПИ им.М.И.Калинина, 195251, Ленинград, Политехническая, 29.

Отпечатано на ротапринте ЛПИ им.М.И.Калинина
195251, Ленинград, Политехническая ул., 29

Работа I

ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Цель работы - умение использовать приближенные методы решения алгебраических и трансцендентных уравнений, разрабатывать соответствующие алгоритмы и программировать их на языке Фортран.

I. Общие положения. Приближенное решение алгебраических и трансцендентных уравнений можно разделить на два этапа. На первом этапе производится отделение корней, т.е. область изменения аргумента разделяется на промежутки, в каждом из которых имеется только один корень уравнения; на втором - вычисление корней с помощью какого-либо численного метода.

Рассмотрим приемы алгоритмизации и программирования второго этапа в решении уравнений, предполагая, что уже осуществлено выделение промежутка изменения аргумента, содержащего вычисляемый корень. Будем считать, что корни являются простыми, случай же кратных корней рассмотрим в конце работы.

Итак, пусть известно, что простой корень ξ уравнения

$$f(x) = 0 \qquad (I.1)$$

содержится на промежутке $[a, b]$. Напомним наиболее употребительные численные методы отыскания корней. Это методы половинного деления, простых итераций, Ньютона и его варианты. Все эти методы позволяют строить числовую последовательность $\{x_k\}$, сходящуюся к искомому корню уравнения, т.е. $|x_k - \xi| \to 0$ при $k \to \infty$. Это предельное соотношение, записанное в виде

$$|x_k - \xi| < \varepsilon, \quad \varepsilon > 0, \qquad (I.2)$$

может служить для оценки близости x_k к корню уравнения. Однако тот факт, что сам корень ξ неизвестен, делает условие (I.2) непригодным. Обычно поступают так. Для левой части неравенства справедлива оценка, следующая из теоремы Лагранжа с учетом того, что $f(\xi) = 0$:

$$|x_k - \xi| \leq |f(x_k)| / m_1,$$

где $m_1 = \min_{x \in [a,b]} |f(x)|$. Отсюда найдем условие, удобное для проверки заданной неравенством (I.2) близости x_k к корню:

$$|f(x_k)| \leqslant m_1 \varepsilon \qquad (I.3)$$

2. Метод половинного деления. Будем считать, что ни один из концов промежутка $[a,b]$ не является корнем уравнения. Методом половинного деления строятся две сходящиеся к корню числовые последовательности $\{\xi_k\}$ и $\{\eta_k\}$. Алгоритм построения их можно описать так. Пусть известны $\xi_k, \eta_k \in [a,b]$ и $f(\xi_k) f(\eta_k) < 0$.

1. Вычислим $c = (\xi_k + \eta_k)/2$ и $f(c)$; если $f(c) = 0$, то c – корень уравнения и вычисления закончены.

2. Вычислим $f(\xi_k) f(c)$, если $f(\xi_k) f(c) < 0$, то положим $\xi_{k+1} = \xi_k$, $\eta_{k+1} = c$. Если же $f(\xi_k) f(c) > 0$, то $\xi_{k+1} = c$, $\eta_{k+1} = \eta_k$.

3. Искомый корень ξ находится на промежутке $[\xi_{k+1}, \eta_{k+1}]$. Если допустимая погрешность ε удовлетворяет неравенству

$$\varepsilon \geqslant \frac{\eta_{k+1} - \xi_{k+1}}{2},$$

то вычисления закончены, в качестве корня следует взять ξ_{k+1} или η_{k+1}. Если предыдущее неравенство не выполняется, то вычисления следует повторить с I.

Для начала счета по этому алгоритму возьмем в качестве ξ_0 и η_0 границы промежутка $[a,b]$, т.е. примем $\xi_0 = a$, $\eta_0 = b$.

Скорость сходимости метода половинного деления определяется неравенством

$$|\xi - x_n| < \frac{b-a}{2^{n+1}}, \quad x_n = \frac{\xi_n + \eta_n}{2}.$$

Последовательности $\{\xi_k\}$, $\{\eta_k\}$ в этом методе сходятся к точке, в которой $f(x)$ меняет знак. Если $f(x)$ непрерывна на $[a,b]$, то такой точкой является корень уравнения (I.I).

На рис. I.I приведена схема, соответствующая этому алгоритму.

3. Метод простых итераций. Здесь предполагается, что уравнение (I.I) приведено тождественным преобразованием к виду

$$x = \varphi(x) \qquad (I.4)$$

Последовательные приближения к корню уравнения вычисляются по формуле

$$x_{k+1} = \varphi(x_k), \quad k = 0, 1, \ldots \qquad (I.5)$$

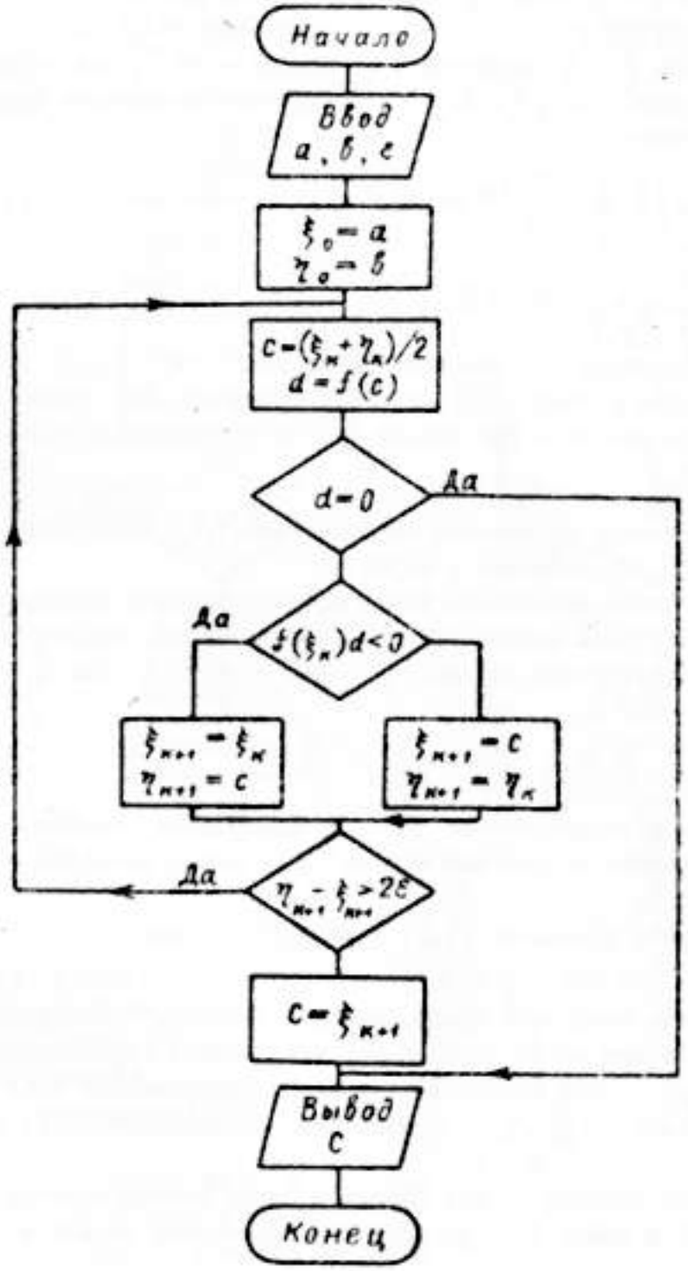

Рис. 1.1

Т е о р е м а I. Если φ , φ' непрерывны на $[a,b]$ и $\varphi(x)\in[a,b]$, $|\varphi'(x)|<1$ для всех $x\in[a,b]$, то последовательность $\{x_k\}$ сходится к ξ при $k\to\infty$, при любом начальном приближении $x_0\in[a,b]$. Скорость сходимости определяется неравенством

$$|\xi - x_{k+1}| \leq \frac{q}{1-q}|x_{k+1} - x_k|, \qquad (I.6)$$

где

$$q = \max_{x\in[a,b]} |\varphi'(x)| .$$

Неравенство (I.6) может служить для оценки близости x_{k+1} к корню и является аналогом условий (I.2) и (I.3). Замечая, что правая часть неравенства (I.6) известна при каждом k , и удовлетворяя условию

$$\frac{q}{1-q}|x_{k+1} - x_k| < \varepsilon ,$$

мы, очевидно, добьемся выполнения неравенства (I.2), т.е. получим требуемую близость приближения к корню.

Запишем алгоритм вычисления корня методом простых итераций. Пусть задана допустимая ошибка ε вычисления корня, некоторое приближение x_k и построена оценка q для $|\varphi'(x)|$ на $[a,b]$.

1. Вычислим $x_{k+1} = \varphi(x_k)$.
2. Вычислим $\frac{q}{1-q}|x_{k+1} - x_k|$.
3. Если эта величина меньше ε , то вычисления закончены и x_{k+1} можно принять за искомый корень ξ , иначе вычисления следует повторить с I.

Для приведения уравнения (I.I) к виду (I.4), где

$$|\varphi'(x)| \leq q < 1, \qquad (I.7)$$

можно использовать следующий прием. Выберем число r и отрезок $[a,b]$ так, чтобы при всех $x\in[a,b]$ выполнялись неравенства $-2+p<rf'<-p$ для P, удовлетворяющего неравенствам $0<p<1$. Положим $\varphi(x) = x + rf(x)$, очевидно, соотношения (I.7) удовлетворяются для $q = 1-p$.

В заключение данного пункта приведем схему метода простых итераций (Рис. I.2) и табл. I.I, связывающую переменные задачи и программы

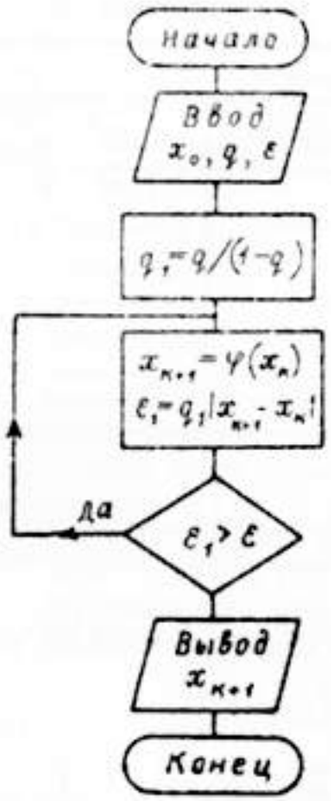

Рис. 1.2

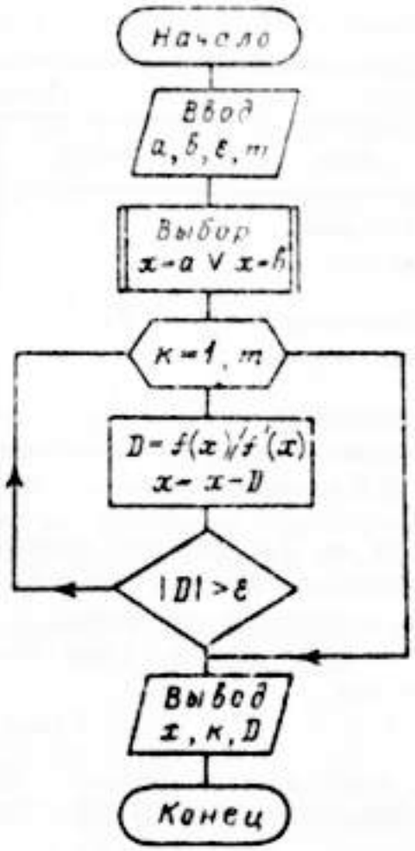

Рис. 1.4

```
      F(X)=...
      READ(5) X,Q,E
      Q1=Q/(Q-1)
    1 Y=F(X)
      E1=Q1*ABS(Y-X)
      X=Y
      IF (E1>E) GO TO 1
      WRITE(7,2) X
    2 FORMAT(' ЗНАЧЕНИЕ КОРНЯ X=',E12.5)
      END RK
```

РИС. 1.3

Таблица I.I

Переменные	Место и тип		
	в задаче	в программе	тип*
Исходные данные	x_0	X	B
	q	Q	— " —
	ε	E	— " —
Вспомогательные величины	ε_1	E1	— " —
	x_1	X1	— " —
	q_1	Q1	— " —
Результат	x_{k+1}	Y	— " —

* B - вещественный.

На рис.I.3 приведена программа вычисления корня методом простых итераций. Отметим, что вид оператор-функции $F(x)$ определяется видом конкретного уравнения.

4. Метод Ньютона (касательных, линеаризации). Формула метода имеет вид

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad n = 0, 1, \ldots \tag{I.8}$$

и может рассматриваться как метод итераций для уравнения (I.4):

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

Имеет место следующая теорема о достаточных условиях сходимости метода Ньютона (I.8) к корню уравнения.

Т е о р е м а 2. Если функция $f(x)$ принимает на концах интервала $[a,b]$ разные знаки, причем f' и f'' непрерывны и знакопостоянны при $x \in [a,b]$, то, выбрав начальное приближение $x_0 \in [a,b]$, удовлетворяющее условию

$$f''(x_0) f(x_0) > 0, \tag{I.9}$$

можно вычислить методом Ньютона (I.8) единственный корень уравнения $f(x) = 0$ с любой степенью точности.

Скорость сходимости метода Ньютона определяется неравенством

$$|\xi - x_{k+1}| \leqslant \frac{M_2}{2m_1} (\xi - x_k)^2, \tag{I.10}$$

где $M_2 = \max|f''(x)|$, $m_1 = \min|f'(x)|$. Формула (I.10) показывает, что метод Ньютона весьма быстро сходится к корню уравнения, сходимость вида (I.10) является квадратичной, т.е. на каждой итерации ошибка $|\xi - x_k|$ возводится в квадрат. При $\xi - x_k \to 0$ число правильных десятичных цифр на каждом шаге примерно удваивается. Пусть, например, $M_2/(2m_1) \approx 1$, тогда при близких ξ и x_k, таких, что $|\xi - x_k| < 1/2$, имеем

$$|\xi - x_{k+1}| < 1/2^2, \quad |\xi - x_{k+2}| < 1/2^4, \ldots$$

$$|\xi - x_{k+6}| < 1/2^{64} \approx 10^{-19}$$

Вспоминая, что количество значащих цифр мантиссы при вычислении с удвоенной точностью в ЭВМ "Искра-1256" равно 12, видим, что, грубо говоря, нужно около шести итераций для сокращения ошибки от 1/2 до величины, меньшей 10^{-12}. Заметим, что для достижения точности методом половинного деления нам пришлось бы сделать примерно 45 итераций $(2^{-39} < 10^{-12} < 2^{-40})$!

Для организации вычислений соотношение (I.10) непригодно и поэтому пользуются условием

$$|\xi - x_{n+1}| \leqslant \frac{M_2}{2m_1} (x_{n+1} - x_n)^2. \tag{I.11}$$

Запишем алгоритм вычисления корня по методу Ньютона. Пусть задана допустимая ошибка ε вычисления корня и найдена величина $M_2/(2m_1)$.

1. Выберем x_0 исходя из условия (I.9), это, например, один из концов промежутка $[a,b]$.

2. Вычислим $x_{k+1} = x_k - f(x_k)/f'(x_k)$.

3. Проверяем неравенство (I.11), записанное в виде

$$\frac{M_2}{2m_1} |x_{k+1} - x_k|^2 \leqslant \varepsilon.$$

4. Если неравенство выполнено, то вычисления закончены и x_{k+1} можно принять за искомый корень, так как $|x_{k+1} - \xi| \leqslant \varepsilon$, иначе вычисления следует повторить с 2.

На рис. I.4. приведена схема метода Ньютона.

В алгоритме введен дополнительный параметр m, используемый для ограничения числа итераций. В результате работы программы выводятся: x - приближенное значение корня, k - число итераций, D - оценка невязки по соотношению типа (I.3).

5. Модифицированный метод Ньютона. Формула метода имеет вид

$$x_{n+1}=x_n-f(x_n)/f'(x_0)\ ,\quad n=0,1,2,\dots \qquad (I.12)$$

где $f'(x_0)\neq 0$. Формула (I.12) совпадает с (I.8) при условии, что на каждом шаге используется $f'(x_n)=f'(x_0)$.

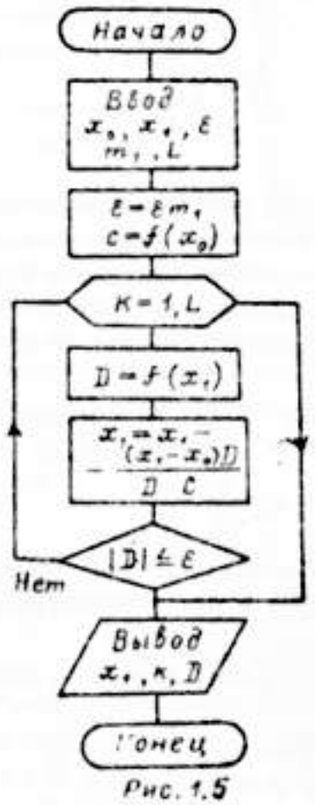

Рис. 1.5

Если рассматривать соотношение (I.12) как вариант метода простой итерации с $\varphi(x)=x-f(x)/f'(x_0)$, то достаточное условие сходимости имеет вид

$$2>f'(x)/f'(x_0)>0. \qquad (I.13)$$

Очевидно, что при условии (I.13) последовательность $\{x_n\}$ может быть немонотонной. Приведем вариант метода, дающий монотонную последовательность $\{x_n\}$, при условии, что $f(x)$ монотонна на $[a,b]$:

$$x_{n+1}=x_n-f(x_n)/M_1, \qquad (I.14)$$

где $n=0,1,2,\dots$

$$M_1=sign(f'(a))\max_{x\in[a,b]}|f'(x)|$$

Поскольку для (I.14) выполняется оценка

$$|x_{n+1}-\xi|\le|x_n-\xi|\,|1-f(\eta)/M_1|\le |x_n-\xi|\ , \qquad (I.15)$$

то метод (I.14) всегда сходится, в частности, и при выполнении условий теоремы I, однако скорость сходимости падает с ростом M_1 , она может быть оценена из соотношения

$$|x_{n+1}-\xi|\le\frac{(b-a)M_2}{M_1}|x_n-\xi|\ . \qquad (I.16)$$

Алгоритм метода аналогичен алгоритму метода касательных. Для оценки попадания в ε - окрестность корня рекомендуется соотношение (I.3).

6. Метод секущих. Формула имеет вид

$$x_{n+1}=x_n-\frac{x_n-x_{n-1}}{f(x_n)-f(x_{n-1})}f(x_n)\ ,\quad n=1,2,3,\dots \qquad (I.17)$$

Сравнивая (I.17) с (I.8), видим, что фактически произошла замена производной $f'(x_n)$ ее конечно-разностным представлением в узлах x_n, x_{n-1} . Такая замена приводит к тому, что скорость сходимости метода секущих стала ниже, чем у метода Ньютона, и может быть оценена из соотношения

$$|x_{n+1}-\xi|\le\left(\frac{M_2}{2m_1}\right)^{\frac{1}{\beta}}|x-\xi|^{\beta}\ , \qquad (I.18)$$

где $\beta=(1+\sqrt{5})/2\simeq 1.62$.

При выводе оценки (I.18) использовано разложение в ряд Тейлора и неравенство (I.3). Последовательность $\{x_n\}$, построенная по (I.17), при выполнении достаточных условий сходится к искомому корню ξ .

Достаточные условия, аналогичные методу простой итерации, приводят к соотношению

$$2>f'(\eta)/f'(\varkappa)>0, \qquad (I.19)$$

где $\eta\in[\xi,x_n]$, $\varkappa\in[x_n,x_{n+1}]$, $n=1,2,\dots$, т.е. $f(x)$ должна быть монотонной на промежутке $[a,b]$, а $f'(x)$ плавно меняется. При этих достаточных условиях последовательность $\{x_n\}$ может быть немонотонной.

Если для $f(x)$ выполняются условия теоремы I, то, выбирая x_0 и x_1 , так, чтобы $(\xi-x_1)/(\xi-x_0)<1$, по формуле (I.17) получим монотонную последовательность $\{x_n\}$, сходящуюся к корню ξ .

Для оценки попадания в ε - окрестность корня можно использовать с отношение

$$\frac{M_2}{2m_1}|x_{n+1}-x_n|\,|x_n-x_{n-1}|<\varepsilon. \qquad (I.20)$$

Однако более простым представляется соотношение (I.3). После выбора начальных значений x_0 , x_1 алгоритм метода аналогичен методу Ньютона.

7. Метод хорд. Основное рекуррентное соотношение метода хорд получается аналогично предыдущим методам и имеет вид

$$x_{n+1}=x_n-\frac{x_n-x_0}{f(x_n)-f(x_0)}f(x_n)\ . \qquad (I.21)$$

По аналогии с модифицированным методом Ньютона здесь в сравнении с методом секущих зафиксировано значение $x_{n-1}=x_0$.

Если $f(x)$ удовлетворяет условиям теоремы I и соответствующим образом выбрано x_0, то последовательность $\{x_n\}$, начиная с x_1, будет монотонно стремиться к ξ . Заметим, что в этом методе, как и в других, приведенных в этой работе, можно сформулировать более слабые достаточные условия, позволяющие построить немонотонную $\{x_n\}$. Скорость сходимости метода хорд может быть оценена неравенством

$$|\xi - x_{n+1}| \le \frac{(l-a)M_2}{2m_1}|\xi - x_n|, \qquad (I.22)$$

т.е. она ниже, чем в методе касательных и секущих.

Для оценки попадания в ε - окрестность корня можно использовать соотношение

$$\frac{(\varepsilon-a)M_2}{2m_1}|x_{n+1}-x_n| < \varepsilon. \qquad (I.23)$$

можно воспользоваться и неравенством (I.3).

П р и м е р. Составить программу вычисления приближенного значения второго положительного корня уравнения

$$x = tgx \qquad (I.24)$$

с помощью метода хорд. Перепишем исходное уравнение в виде

$$f(x) = -x\cos x + \sin x = 0. \qquad (I.25)$$

Заметим, что $f'(x)= x\sin x$ и $f''(x)= \sin x + x\cos x$ на интервале $\left[\frac{5\pi}{4}, \frac{3\pi}{2}\right]$, где находится искомый корень, отрицательны, а $f(x)$ монотонна, т.е. выполнены условия теоремы I. Можно показать, что $m_1 = \frac{5\pi}{4\sqrt{2}}$, а $x_0=\frac{3\pi}{2}$, $x_1=\frac{5\pi}{4}$.

На рис. I.5. изображена схема программы.

В схеме введено ограничение на число итераций L , что необходимо для подсчета числа итераций и устранения возможного "зацикливания" программы при несоблюдении достаточных условий сходимости.

```
PROGRAM  F(X)=-X*COS(X)+SIN(X)
READ(5) X0,X1,EP,AM1,L
EP=EP*AM1 C=F(X0)
DO 1 K=1,L
D=F(X1) X1=X1-(X1-X0)*D/(D-C)
IF (ABS(D)<EP) GO TO 2
1 CONTINUE
2 WRITE(7,3) K,X1,D
3 FORMAT(I3,2E12.5)
 STOP   END XM
```

Рис. 1.6

Таблица I.2

Переменные	Место и тип		
	в задаче	в программе	
Исходные данные	x_0	X0	B
	x_1	X1	—"—
	ε	EP	—"—
	m_1	AM1	—"—
Вспомогательные величины	$f(x_n)$	D	—"—
	$f(x_0)$	C	—"—
Результат	L, K	L, K	u*
	x_{n+1}	X1	B

* u - целый.

На рис. I.6 приведена программа для ЭВМ "Искра-I256", соответствие переменных программы и задачи указано в табл. I.2.

В случае вычисления кратного корня функции не все рассмотренные выше методы могут оказаться применимыми либо оказываются малоэффективными. Так, если вычисляемый корень имеет четную кратность, то $f(x)$ не меняет знака в точке ξ , поэтому метод половинного деления неприменим. Указанный выше способ приведения уравнения (I.I) к "итеративному" виду (I.4) также не дает результатов. Для вычисления корня нечетной кратности эти методы применимы, но их эффективность падает из-за быстрого убывания значений функции $f(x)$ в окрестности нуля.

Простой способ сохранить эффективность обоих методов состоит в использовании производной $f^{(m)}(x)$ вместо функции $f(x)$, где m - кратность корня. В тех случаях, когда вычисление производной затруднительно, можно использовать функцию $[f(x)]^{\frac{1}{m}}$ вместо $f(x)$, однако при четной кратности m для вычисления корня функции $[f(x)]^{\frac{1}{m}}$ рассмотренные выше методы половинного деления и итераций неприменимы. Здесь оказывается эффективным метод Ньютона и его варианты.

З а д а н и я. В задачах 1-30 требуется с помощью указанных методов найти с точностью ε корень (корни) уравнения. Если корней несколько, необходимо предварительно провести отделение корней.

1. Найти корень уравнения методом итерации. В тех случаях, когда указан интервал отделения корня, нужно разыскивать корень из этого интервала. Когда это необходимо, уравнения должны быть приведены к виду, пригодному для применения метода итерации:

1) $3x + lg(3x+2) = 1$;
2) $x = e^{-x^2} + 1$;
3) $x = \cos x$;
4) $x^3 - 2x^2 + 7x + 3 = 0$, $\xi \in [-1,0]$;
5) $x^3 - 2x^2 - 4x + 7 = 0$, $\xi \in [-2/3, 2]$;
6) $x^4 - 4x^3 + 4x^2 - 4 = 0$, $\xi \in [-1,0]$;
7) $x^4 - 10x^2 - 16x + 5 = 0$, $\xi \in [0,1]$;
8) $x^4 - 6x^2 + 12x - 8 = 0$, $\xi \in [1,2]$;
9) $x^4 - x^3 - 7x^2 - 8x - 6 = 0$, $\xi \in [3,4]$;
10) $x^4 - 3x^3 + 3x^2 - 2 = 0$, $\xi \in [1,2]$.

2. Найти корень (корни) уравнения, используя либо метод Ньютона, либо метод секущих, либо метод хорд:

11) $x^2 - 10\,lg x - 3 = 0$;
12) $\sin x + x - 1 = 0$;
13) $(x-1)^2 - e^{-x} = 0$;
14) $x^3 - 3x^2 - 3x + 11 = 0$;
15) $x^3 - 2x^2 - 4x - 7 = 0$;
16) $x^4 - 3x^2 + 75x - 9999 = 0$;
17) $x^4 + 3x^3 - 9x - 9 = 0$, $\xi \in [1,2]$;
18) $x^4 + 3x^3 + 4x^2 + x - 3 = 0$, $\xi \in [0,1]$;
19) $x^4 - x^3 - 2x^2 + 3x - 3 = 0$, $\xi \in [-2,-1]$;
20) $x^5 + 5x + 1 = 0$. $\xi \in [-2,0]$;

3. Найти корень уравнения одним из приведенных в работе методов:

21) $3lg^2 x + 6ln(1+x) - 3 = 0$, $\xi \in [0.1,3]$;
22) $2x\sin x - \cos x = 0$, $\xi \in [0.4,1]$;
23) $0.25x^3 + x - 1.2502 = 0$, $\xi \in [0,2]$;
24) $0.1x^2 - x\ln x = 0$, $\xi \in [1,2]$;
25) $3x - 4\ln x - 5 = 0$, $\xi \in [2,4]$;

26) $e^x - e^{-x} - 2 = 0$, $\xi \in [0,1]$;
27) $e^x + \ln x - 10x = 0$, $\xi \in [3,4]$;
28) $3x - 14 + e^x - e^{-x} = 0$, $\xi \in [1,3]$;
29) $x\,tg\,x - \frac{1}{3} = 0$, $\xi \in [0,2,1]$;
30) $\sqrt{1-x} - \cos\sqrt{1-x} = 0$ $\xi \in [0,1]$.

Р а б о т а 2

ИНТЕРПОЛЯЦИОННЫЕ ПОЛИНОМЫ ЛАГРАНЖА И НЬЮТОНА. СПЛАЙНЫ

Цель работы - умение применять интерполяционные полиномы в программах, использующих методы приближения функций.

1. Общие положения. При использовании ЭВМ приходится иметь дело с табличными функциями, что обусловлено как особенностями вычислительной математики, так и способами представления данных в цифровых вычислительных машинах. Таблица является совокупностью n пар чисел $(x_i, y_i), i = 1,2,\ldots,n$, где x_i - значения аргумента (узлы), а $y_i = y(x_i)$ - значения функции в узлах. Однако на практике требуется знать значения функции $y(x)$ в точках, не совпадающих с узлами. Существуют различные способы восстановления функции по ее узловым значениям. Одной из основных идей является следующая. Выбирают систему некоторых функций $\varphi_j(x)$, $j = 1,\ldots,m$, например, $\varphi_j(x) = x^{j-1}$. Находят числа $\alpha_j, j = 1,\ldots,m$, так, чтобы функция

$$\varphi(x) = \sum_{j=1}^{m} \alpha_j \varphi_j(x) \qquad (2.1)$$

была бы близка в некотором смысле к заданной функции $y(x)$. При интерполировании предполагается, что $n = m$, а условие близости функций $\varphi(x)$ и $y(x)$ сводится к совпадению их значений в точках x_i:

$$\varphi(x_i) = y_i \quad , \quad i = 1,2,\ldots,n. \qquad (2.2)$$

Если $n = m$, то количество неизвестных α_j равно числу уравнений (2.2), содержащих эти числа. При алгебраическом интерполировании, т.е. когда

$$\varphi_j(x) = x^{j-1} \quad , \quad j = 1,2,\ldots,n, \qquad (2.3)$$

для однозначной разрешимости системы линейных уравнений (2.2) отно-

сительно чисел α_j необходимо и достаточно, чтобы все узлы x_j были различны. Полученная функция $\varphi(x)$ будет полиномом степени $n-1$ и называется интерполяционным полиномом. Хотя этот полином является единственным при отмеченных условиях, существуют различные формы его записи.

2. Интерполяционный полином в форме Лагранжа. Этот полином будем обозначать $L_{n-1}(x)$. Он может быть записан в виде

$$L_{n-1}(x)=\sum_{k=1}^{n} y_k \prod_{\substack{i=1\\ i\neq k}}^{n} \frac{x-x_i}{x_k-x_i} \qquad (2.4)$$

Если ввести вспомогательный полином

$$\omega_n(x)=\prod_{i=1}^{n}(x-x_i), \qquad (2.5)$$

то формулу (2.4) можно переписать в форме

$$L_{n-1}(x)=\sum_{k=1}^{n} y_k \frac{\omega_n(x)}{(x-x_k)\omega_n'(x_k)}. \qquad (2.6)$$

Хотя формула (2.6) чаще встречается в литературе, но при создании программ, по нашему мнению, предпочтение следует отдать формуле (2.4).

Если все узлы являются равноотстоящими, т.е.

$$x_i=a+(i-1)h, \quad i=\overline{1,n}, \qquad (2.7)$$

можно ввести вместо переменной x переменную t по формуле

$$t=(x-a)/h \qquad (2.8)$$

и тогда интерполяционный полином (2.4) примет вид

$$L_{n-1}(x)=L_{n-1}(a+th)=\sum_{k=1}^{n} y_k \prod_{\substack{i=1\\ i\neq k}}^{n} \frac{t-i}{k-i}. \qquad (2.9)$$

3. Интерполяционный полином в форме Ньютона. Этот полином мы будем обозначать $P_{n-1}(x)$. Он представим в виде

$$P_{n-1}(x)=y(x_1)+(x-x_1)\cdot y(x_1,x_2)+\ldots+\prod_{i=1}^{n-1}(x-x_i)\cdot y(x_1,\ldots x_n)= \qquad (2.10)$$
$$=y(x_1)+\sum_{k=2}^{n} y(x_1,\ldots,x_k)\prod_{i=1}^{k-1}(x-x_i).$$

В формуле (2.10) через $y(x_1,\ldots,x_k)$ обозначены разделенные разности. При создании эффективных алгоритмов вычисления полинома $P_{n-1}(x)$ следует пользоваться рекуррентными формулами для разделенных разностей. Однако для упрощения создаваемых программ мы воспользуемся представлением разделенной разности через значения функции в узлах

$$y(x_1,\ldots,x_k)=\sum_{j=1}^{k} y(x_j)\frac{1}{\prod_{\substack{i=1\\ i\neq j}}^{k}(x_j-x_i)}. \qquad (2.11)$$

Отметим, что в различных случаях бывает более удобно применять ту или другую форму интерполяционного полинома. Так, если приходится работать с несколькими функциями, заданными таблично на одних и тех же узлах, то интерполяционный полином в форме Лагранжа бывает предпочтительнее. С другой стороны, если приходится работать с одной табличной функцией, но с переменным числом узлов (например, добавляются новые экспериментальные значения), то предпочтение следует отдать интерполяционному полиному в форме Ньютона.

Погрешность интерполяционного полинома определяется формулой остаточного члена для точек x, не совпадающих с узлами:

$$y(x)-L_{n-1}(x)=r_{n-1}(x)=\frac{\omega(x)}{(n+1)!}y^{(n)}(\xi). \qquad (2.12)$$

В эту формулу входит n-я производная от функции, заданной таблично, причем в некоторой заранее неизвестной точке ξ. Поэтому формула (2.12) имеет чисто теоретическое значение, и мы не будем уделять ей внимание.

В заключение сформулируем некоторые теоремы, относящиеся к вопросу сходимости интерполяционного процесса, т.е. к поведению интерполяционных полиномов при увеличении числа узлов.

Т е о р е м а I. Для каждой непрерывной на конечном интервале $[a,b]$ функции $f(x)$ существуют такие множества узлов

$$\left\{x_i^{(n)}\right\}, \quad i=\overline{1,n}; \quad n=1,2,\ldots,$$

что последовательность интерполяционных полиномов $\left[L_{n-1}(x)\right]$, построенная по каждому набору узлов, равномерно сходится к $f(x)$ на $[a,b]$.

Отметим, что для каждой функции $f(x)$ множества узлов, о которых идет речь в теореме I, могут быть свои.

Т е о р е м а 2. Не существует таких множеств узлов $\{x_i^{(n)}\}$ $i=1,2,\dots,n$; $n=1,2,\dots$ на конечном интервале $[a,b]$, что для любой непрерывной на $[a,b]$ функции $f(x)$ последовательность интерполяционных полиномов $\{L_{n-1}(x)\}$, построенных по этим узлам, равномерно сходятся к функции $f(x)$.

Таким образом, даже для непрерывной функции увеличение количества узлов не гарантирует ее лучшего приближения с помощью интерполяционного полинома.

П р и м е р. По таблично заданной функции построить

i	1	2	3	4	5
x_i	0	0,5	1	1,5	2
y_i	0	1	0	-1	0

интерполяционный полином в форме Лагранжа и сравнить его значение в заданной точке X со значением функции $y=\sin \pi x$, совпадающей в узлах с узловыми значениями табличной функции. Составить алгоритм решения поставленной задачи.

Р е ш е н и е. Для построения интерполяционного полинома все узлы должны быть различными. Это условие выполнено.

Схема программы приведена на рис. 2.1. В схеме введены следующие обозначения:

$$P=\prod_{\substack{i=1\\ i\neq k}}^{n}\frac{x-x_i}{x_k-x_i}, \qquad YL=\sum_{k=1}^{n} y_k \prod_{\substack{i=1\\ i\neq k}}^{n}\frac{x-x_i}{x_k-x_i}$$

Кратко поясним содержание схемы. Алгоритм вычисления интерполяционного полинома в форме Лагранжа в заданной точке является двойным циклическим процессом. Внутренний цикл (блоки 5,6,7,8) обеспечивает вычисление величины $\prod_{\substack{i=1\\ i\neq k}}^{n}(x-x_i)/(x_k-x_i)$. Внешний цикл (блоки 3-9) обеспечивает вычисление интерполяционного полинома по формуле (2.4).

4. Интерполяция кубическими сплайнами. На больших промежутках (при больших n) применение интерполяционных полиномов Лагранжа или Ньютона неудобно по следующим причинам.

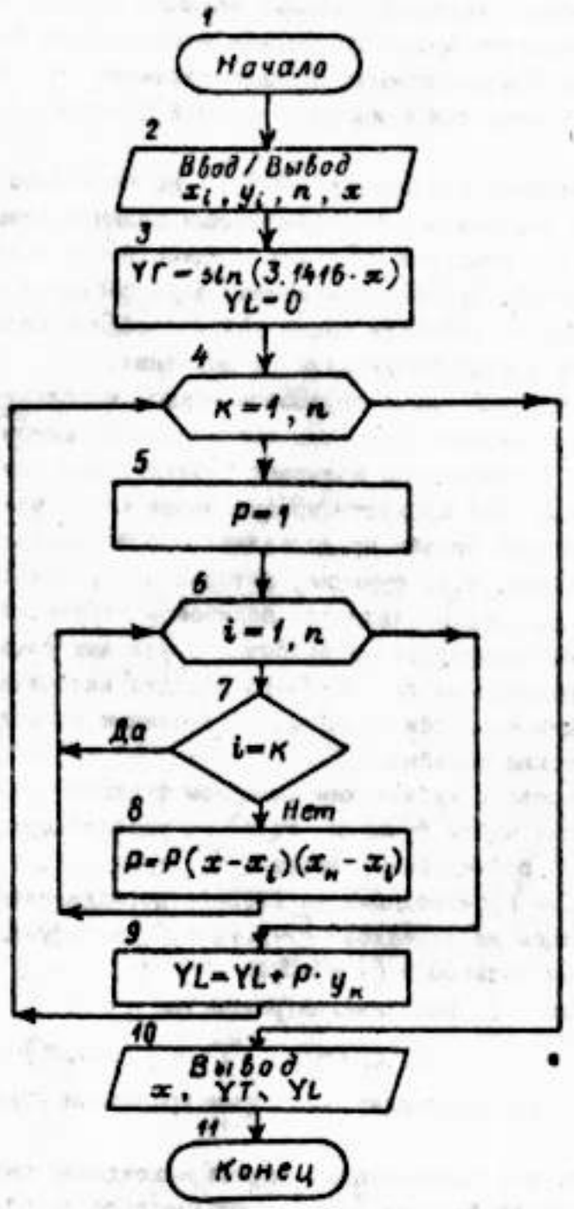

Рис. 2.1

I. Если для достижения нужной точности использовать много узлов, т.е. применить интерполяционный полином высокой степени, то значительно возрастет время построения и вычисления полинома; кроме того, не всегда при увеличении порядка полинома n последовательность полиномов сходится к интерполируемой функции $y(x)$ (см. теоремы I и 2).

2. Если разбить промежуток $[a,b]$ на несколько участков и на каждом участке построить интерполяционный полином невысокой степени, то на стыках этих участков не будет гладкости (поскольку, вообще говоря, терпит разрыв первая производная двух соседних полиномов), а если точка стыка не является общим узлом соседних полиномов, то могут не совпасть в этой точке даже их значения.

Избежать перечисленные трудности можно, используя приближения сплайнами вместо интерполяционных полиномов. Сплайном ($spline$ - линейка (англ.)) чертежники называют гибкую рейку, которую применяют как лекало, если надо провести кривую через много узлов.

Математический сплайн предоставляет собой гладкую кусочно-полиномиальную функцию, т.е. функцию, которая между каждой соседней парой узлов $[x_{i-1}, x_i]$ является полиномом некоторой степени m, причем полиномы соединяются непрерывно в узловых точках x_i, так же как и их производные до $(m-1)$-го порядка включительно. Наиболее употребительными являются сплайны, построенные из полиномов 3-й степени, - кубические сплайны.

Интерполировать кубическим сплайном функцию $y(x)$, заданную таблично, значит найти функцию $S(x)$, удовлетворяющую требованиям:

I) $S(x)$ принадлежит классу $C^{(2)}[a,b]$, т.е. непрерывна вместе со своими производными до второго порядка включительно;

2) на каждом из отрезков $[x_{i-1}, x_i]$ функция $S(x)$ является кубическим полиномом $(i=2,3,\dots,n)$;

3) в узлах x_i выполняются равенства

$$S(x_i) = y_i \quad (i=1,2,\dots,n); \tag{2.I3}$$

4) $S(x)$ удовлетворяет некоторым граничным условиям в узлах a и b.

Доказано, что поставленная задача нахождения интерполирующей кусочно-кубической функции $S(x)$ (кубического сплайна) имеет единственное решение.

Расчетные формулы. Так как вторая производная функции $S(x)$ непрерывна и линейна на каждом отрезке $[x_{i-1}, x_i]\,(i=2,3,\dots,n)$, ее

представляют при $x_{i-1} \le x \le x_i$ в виде

$$S''(x) = m_{i-1}\frac{x_i - x}{h_i} + m_i\frac{x - x_{i-1}}{h_i}, \tag{2.I4}$$

где $h_i = x_i - x_{i-1}$, $m_k = S''(x_k)$. Интегрируя дважды обе части равенства (2.I4) и используя условия (2.I3), получают выражения для $S(x)$ и $S'(x)$ на отрезке $[x_{i-1}, x_i]$:

$$S(x) = m_{i-1}\frac{(x_i - x)^3}{6h_i} + m_i\frac{(x - x_{i-1})^3}{6h_i} + \tag{2.I5}$$

$$+\left(y_{i-1} - \frac{m_{i-1}h_i^2}{6}\right)\frac{x_i - x}{h_i} + \left(y_i - \frac{m_i h_i^2}{6}\right)\frac{x - x_{i-1}}{h_i} \quad (i=2,3,\dots,n-1);$$

$$S'(x) = -m_{i-1}\frac{(x_i - x)^2}{2h_i} + m_i\frac{(x - x_{i-1})^2}{2h_i} + \frac{y_i - y_{i-1}}{h_i} - \tag{2.I6}$$

$$-\frac{m_i - m_{i-1}}{6}h_i \qquad (i=2,3,\dots,n-1).$$

Итак, для определения сплайна $S(x)$ на $[a,b]$ надо вычислить величины $m_1, m_2, \dots, m_n$. Из условия непрерывности $S'(x)$ в узлах $x_2, x_3, \dots, x_{n-1}$ записываются $(n-2)$ уравнения:

$$\frac{h_i}{6}m_{i-1} + \frac{h_i + h_{i+1}}{3}m_i + \frac{h_{i+1}}{6}m_{i+1} = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \quad (i=2,3,\dots,n-1). \tag{2.I7}$$

Дополнив эти уравнения граничными условиями, получают систему n уравнений для определения n неизвестных $m_1, m_2, \dots, m_n$.

Рассмотрим различные случаи задания граничных условий.

I. Заданы условия "свободного провисания" интерполяционной кривой в точках a и b ("естественный сплайн"):

$$S''(a) = S''(b) = 0. \tag{2.I8}$$

Дополнив уравнения (2.I7) равенствами $m_1 = m_n = 0$, получаем систему

$$Am = Hy \tag{2.I9}$$

Квадратная матрица A имеет вид:

$$A=\begin{pmatrix} \frac{h_2+h_3}{3} & \frac{h_3}{6} & 0 & \dots & 0 & 0 \\ \frac{h_3}{6} & \frac{h_3+h_4}{3} & \frac{h_4}{6} & \dots & 0 & 0 \\ 0 & \frac{h_4}{6} & \frac{h_4+h_5}{3} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \frac{h_{n-1}}{6} & \frac{h_{n-1}+h_n}{3} \end{pmatrix} \qquad (2.20)$$

векторы m , y и прямоугольная матрица H :

$$m=\begin{pmatrix} m_2 \\ m_3 \\ \vdots \\ m_{n-1} \end{pmatrix}; \qquad y=\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad (2.21)$$

$$H=\begin{pmatrix} \frac{1}{h_2} & \left(-\frac{1}{h_2}-\frac{1}{h_3}\right) & \frac{1}{h_3} & 0\dots0 & 0 \\ 0 & \frac{1}{h_3} & \left(-\frac{1}{h_3}-\frac{1}{h_4}\right) & \frac{1}{h_4}\cdots0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0\dots\left(-\frac{1}{h_{n-1}}-\frac{1}{h_n}\right) & \frac{1}{h_n} \end{pmatrix}$$

Матрица A симметричная, трехдиагональная, со строгим диагональным преобладанием, поэтому величины $m_2,\dots, m_{n-1}$ определяются однозначно, а задачао нахождении сплайна $S(x)$ имеет единственное решение (как и было указано выше). Систему (2.I9) удобно решать методом прогонки, специально предназначенным для решения систем алгебраических уравнений с трехдиагональной матрицей (см. лабораторную работу по прямым методам решения линейных уравнений).

Далее, зная величины $m_1, m_2,\dots,m_n$, можно по формуле (2.I5) вычислить значение сплайна в любой точке $z \in [a,b]$, предварительно определив отрезок $[x_i, x_{i+1}]$, которому принадлежит z : $x_i \le z \le x_{i+1}$.

2. Известны наклоны интерполяционной кривой в граничных точках:

$$S'(a)=y'_1 , \quad S'(b)=y'_n . \qquad (2.22)$$

Используя формулу (2.I), получают равенства

$$\frac{2}{3}m_1+\frac{1}{3}m_2=\frac{2}{h_2}\left(\frac{y_2-y_1}{h_2}-y'_1\right); \qquad (2.23)$$
$$\frac{1}{3}m_{n-1}+\frac{2}{3}m_n=\frac{2}{h_n}\left(y'_n-\frac{y_n-y_{n-1}}{h_n}\right),$$

пополняют ими совокупность уравнений (2.I7) и решают полученную систему n уравнений с n неизвестными аналогично случаю I.

3. Если известна кривизна $y(x)$ в граничных точках

$$S''(a)=y''_1 , \quad S''(b)=y''_n , \qquad (2.24)$$

то система уравнений (2.I7) пополнится равенствами

$$m_1=y''_1 \qquad m_n=y''_n . \qquad (2.25)$$

Далее систему решают аналогично случаю I.

Возможны другие граничные условия и комбинации рассмотренных условий в узлах a и b .

Если значения сплайна нужно находить многократно или выполнять некоторые операции с ним, например, вычислять производные, интегралы, то лучше иметь сплайн в виде набора коэффициентов кубических полиномов

$$S(x)=y_i+a_i(x-x_i)+b_i(x-x_i)^2+c_i(x-x_i)^3, \qquad (2.26)$$

где $x_i \le x \le x_{i+1}$ $(i=1,2,\dots,n-1)$

Записав (2.I5) для отрезка $[x_i, x_{i+1}]$ и перегруппировав члены, выделяя коэффициенты при соответствующих степенях $(x-x_i)$, получим

$$a_i=\frac{y_{i+1}-y_i}{h_i}-\frac{h}{6}(2m_i-m_{i+1}); \quad b_i=\frac{m_i}{2} , \qquad (2.27)$$
$$c_i=\frac{m_{i+1}-m_i}{h_i} \quad (i=1,2,\dots n-1)$$

Зная $m_1, m_2,\dots, m_n$, легко вычислить коэффициенты для любого отрезка $[x_i, x_{i+1}]$.

Кубический сплайн достаточно хорошо приближает гладкие функции вместе с несколькими производными. Для простоты приведем оценку погрешности кубического сплайна на равноотстоящих узлах с $h=(b-a)/n$.

Т е о р е м а 3. Если $y(x) \in C^{(k+1)}[a,b]$, $0 \le k \le 3$, о

то кубический сплайн $S(x)$ удовлетворяет неравенству

$$\max_{[x_i,x_{i+1}]}\left|y^{(m)}(x)-S^{(m)}(x)\right| \le Ch^{K+1-m}\max_{[a,b]}\left|y^{(K+1)}(x)\right|, \qquad (2.28)$$

где $i=1,2,\dots,n$, $m=0,1,\dots,K$, C – не зависящая от h , i , y постоянная.

Таким образом, если $y(x)\in C^{(4)}[a,b]$, т.е. $K=3$, то на $[a,b]$ максимальные по модулю уклонения $S(x)$ от $y(x)$, $S'(x)$ от $y'(x)$ и $S''(x)$ от $y''(x)$ имеют порядок $O(h^4)$, $O(h^3)$ и $O(h^2)$. На точность представления функции $y(x)$ сплайном $S(x)$, как видим, можно влиять выбором n , т.е. выбором шага таблицы $h=(b-a)/n$.

Приведем возможный алгоритм решения на ЭВМ задачи интерполирования функции кубическим сплайном.

I. Формируется система вида $Am=f$ $(f=f_1,f_2,\dots,f_n-$ свободные члены системы) из условия (2.I3), уравнений (2.I7) и граничных условий.

2. Определяются значения $m_1,m_2,\dots,m_n$ как решения системы $Am=f$ (рекомендуется использовать для решения системы метод прогонки).

3. Вычисляется значение сплайна $S(z)$ в заданной точке z .

П р и м е р. Известна таблица I0 значений функции $y(x)$ на промежутке $[0,1]$ и наклоны $y(x)$ к оси X в граничных точках $y'(0)=0$, $y'(1)=0$.

x_i	0.000	0.I25	0.205	0.300	0.380	0.500	0.600	0.690	0.800	I.000
y_i	0.200	0.I88	0.I58	0.I00	0.I20	0.2I4	0.3I2	0.390	0.478	0.500
i	I	2	3	4	5	6	7	8	9	I0

Построить кубический сплайн, интерполирующий функцию $y(x)$ по этой таблице, удовлетворяющий заданным граничным условиям, и вычислить его значения в точках Z = 0,05, 0,20, 0,55, 0,90.

Для решения системы линейных алгебраических уравнений методом прогонки используем алгоритм, схема которого приведена в лабораторной работе по прямым методам решения линейных уравнений.

Рекомендуемый состав программы: главная программа (рис.2.2);

подпрограмма метода прогонки, назовем ее $PROG$; внешняя функция $ZSPL$ вычисления значения сплайна в точке Z (рис. 2.3).

Более глубокая детализация программы зависит от оформления метода прогонки и поэтому здесь не приводится.

При решении задачи на ЭВМ "Искра-I256" получены следующие значения кубического сплайна:

(0.05) = 0.I98I
(0.20) = 0.I6I0
(0.55) = 0.2639
(0.90) = 0.487I

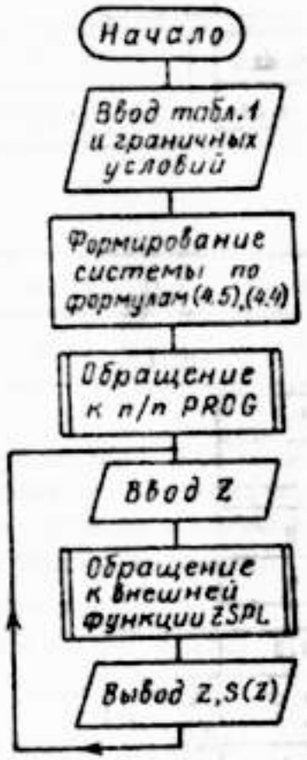

Рис. 2.2

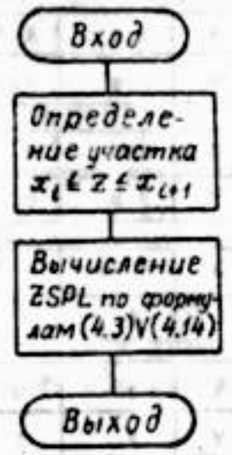

Рис. 2.3

З А Д А Н И Я

I. Построить интерполяционный полином в форме Лагранжа или Ньютона либо приблизить табличную функцию сплайнами. Найти значения табличной функции в точках $0{,}5(X_i + X_{i-1})$
(граничные условия при интерполировании сплайнами задает преподаватель).

1.

X	0.1	0.2	0.3	0.4	0.5
Y	1.3	1.7	1.8	1.65	1.35

2.

X	-0.12	1.68	3.41	5.62
Y	0.324	-0.6	-0.4	-0.21

3.

X	1.1	1.3	1.7
Y	2.1	-2.1	2.1

4.

X	2.0	2.3	2.5
Y	5.8	6.3	6.5

5.

X	1.43	1.47	1.49	1.51
Y	1	2	3	4

6.

X	-1	0	1	2
Y	2	1	2	9

7.

X	0.0	0.1	0.2	0.3	0.4
Y	0.95	1.02	1.11	1.18	1.26

8.

X	0.5	0.6	0.7	0.8
Y	1.32	1.38	1.45	1.51

9.

X	0.0	0.2	0.4	0.6	0.8
Y	1.62	1.65	1.67	1.62	1.79

10.

x_i	0.1	0.3	0.5	0.7
y_i	I.68	I.75	I.79	I.83

11.

x_i	0.0	0.2	0.4	0.6	0.8
y_i	I.83	I.85	I.84	I.82	I.79

12.

x_i	0.0	0.1	0.2	0.3	0.4	0.5
y_i	I.74	I.72	I.70	I.68	I.66	I.64

13.

x_i	0.1	0.3	0.5	0.7
y_i	I.06	I.21	I.35	I.47

II. Дана функция $y(x)$. По заданной таблице узлов построить интерполяционный полином в форме Лагранжа или Ньютона и сравнить точное значение функции и значение интерполяционного полинома в средних точках каждого интервала.

I4. $y(x) = \sin \pi x$
$x_i = 0.1;\quad 0.32;\quad 0.75;\quad 0{,}9;\quad 1{,}2.$

I5. $y(x) = \ln(1 + x^2)$
$x_i = 0.3;\quad 0{,}5;\quad 0{,}7;\quad 0{,}9;\quad 1{,}1.$

I6. $y(x) = 5e^{-x^2}$
$x_i = -0.2;\quad 0.05;\quad 0{,}27;\quad 0{,}43.$

17. $y(x)=ch\,x$

x_i = -0,3; 0,I; 0,9.

18. $y(x)= x^5+4x^3+1$

x_i = 0,25; 0,5; 0,75; I,0.

19. $y(x)= 2\cos x$

x_i = -0.3I; -0,2I; -0,II; -0,0I.

20. $y(x)= \frac{1}{x^2+1}$

x_i = -0,3; -0,I; 0,I; 0,5; 0,7.

21. $y(x)=3\,arctg\,x$

x_i = - 3; -2; 0; 4; I0.

22. $y(x)= 2^{\sin x}$

x_i = 0,I; 0,9; I,5; I,9; 2,5.

23. $y(x)= th\,x$

x_i = 0,3; 0,4; 0,5; 0,6; 0,7; 0.8.

24. $y(x)= ln\,x$

x_i = I,20; I,25; I,30; I,35.

25. $y(x)= \frac{1}{1+25x^2}$

x_i = - I,0; - 0,9; ...; 0,9; I,0 (= 2I).

Работа 3

ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ

Цель работы - умение применять квадратурные формулы трапеций, Симпсона и Гаусса для вычисления интегралов, знание практических приемов оценки погрешности интегрирования и умение программировать эти формулы и приемы на языке Фортран.

I. Общие положения. Формулы приближенного вычисления одномерных интегралов называются квадратурными. Любая квадратурная формула может быть представлена в следующем виде:

$$\mathcal{J} = \int_a^b y(x)\,dx \approx F = \sum_{k=1}^{n} C_k y_k \qquad (3.1)$$

где $\mathcal{J}$ - точное значение интеграла, а F - его приближенное значение, полученное по квадратурной формуле, C_k - коэффициенты; $y_k = y(x_k)$ - значения подынтегральной функции, вычисленные в узлах $x_k \in [a,b]$. Набор C_k и x_k ($k = 1,2, \ldots, n$) определяет конкретную квадратурную формулу.

Для увеличения точности промежуток интерполирования $[a,b]$ разбивают на m участков длиной $h=(b-a)/m$, к каждому участку применяют квадратурную формулу, а затем складывают результаты. Формулы, полученные таким образом, называются обобщенными квадратурными формулами. Приближенное значение интеграла, вычисленное по обобщенной формуле обозначим F_h, так как оно зависит от числа разбиения промежутка m, а следовательно, и от шага интегрирования h.

Разность между точным $\mathcal{J}$ и приближенным F_h значениями интеграла называется погрешностью квадратурной формулы. Она может быть выражена равенством:

$$\mathcal{J} = F_h - h^r \rho_r + O(h^{r+1}), \qquad (3.2)$$

где r - целое число, конкретное значение которого для каждой квадратурной формулы известно, а ρ_r - число, которое определяется через интеграл от производной $y^{(r)}(x)$, если, конечно, эта производная существует и непрерывна.

Точное значение погрешности по соотношению (3.2) вычислить принципиально невозможно, а теоретические оценки этой погрешности как правило получаются завышенными. Поэтому при расчетах на ЭВМ пользуются следующим приемом. Интеграл вычисляют дважды с шагами интегрирования $2h$ и h по полученным результатам F_{2h} и F_h находят приближенное значение погрешности $\mathcal{E}_F$ с помощью формулы Рунге:

$$|\mathcal{J} - F_h| \approx \mathcal{E}_F = \frac{|F_h - F_{2h}|}{2^r - 1}, \qquad (3.3)$$

где r - степень в оценке (3.2).

Рассмотренный прием позволяет решать два типа задач численно-

го интегрирования:

I) вычисление приближенного значения интеграла с заданным числом разбиений промежутка интегрирования m (или с заданным шагом h) и последующим нахождением по формуле (3.3) величины ε_F, которая является оценкой погрешности;

2) вычисление приближенного значения интеграла с заданной точностью ε; здесь также задается начальное число m (или начальный шаг h), а вычисления интеграла производят многократно, каждый раз уменьшая шаг вдвое до тех пор, пока для очередных двух значений F_{2h} и F_h не выполнится условие

$$\varepsilon_F = \frac{|F_h - F_{2h}|}{2^r - 1} < \varepsilon \qquad (3.4)$$

Решение одной из указанных задач с помощью обобщенных квадратурных формул трапеций или Симпсона, а также формулы высшего класса точности Гаусса на ЭВМ "Искра-1256" и является содержанием лабораторной работы.

2. Вычисление интеграла по формуле трапеций и Симпсона. Обобщенную формулу трапеций можно представить в следующей удобной для программирования форме:

$$F_h = h\left(\frac{y_0 + y_m}{2} + y_1 + y_2 + \dots + y_{m-1}\right), \qquad (3.5)$$

где $h = (b-a)/m$ - шаг интегрирования;

$y_k = y(x_k)$ - значения подынтегральной функции в равноотстоящих точках $x_k = a + kh$, $k = 0, 1, \dots, m$

Формула Рунге (3.2) для погрешности интегрирования в случае обобщенной формулы трапеций имеет вид ($r = 2$)

$$\varepsilon_F = \frac{|F_h - F_{2h}|}{3} \qquad (3.6)$$

Рассмотрим применение обобщенной квадратурной формулы трапеций для решения задачи первого типа.

Пример 1. Вычислить интеграл

$$J = \int_a^b \frac{e^x}{1+x}\,dx, \quad a = 1, \quad b = 2.$$

по формуле трапеций при трех вариантах разбиения промежутка интегрирования: $m = 2$, 10 и 30. Найти погрешности полученных результатов.

Программу составим так, чтобы число разбиений $m = 2L$ промежутка $[a,b]$ могло быть произвольным. Это позволит выполнить три варианта расчета для $L = 1, 5, 15$. Следует только предусмотреть в программе ввод L с клавиатуры.

Расчет интеграла будет производиться по формуле (3.5), а расчет погрешности - по формуле (3.6).

Квадратурная формула (3.5) требует многократного вычисления подынтегральной функции $y(x)$, поэтому ее вычислительный алгоритм удобно оформить в виде внешней функции. Для определения ε_F по формуле (3.6) вычисление интеграла придется производить два раза: с числом дроблений $m = L$ (значение F_{2h}) и $m = 2L$ (значение F_h). Поэтому алгоритм вычисления интеграла также следует оформить в виде внешней функции $F(a, b, m, y)$, зависящей от пределов интегрирования, числа разбиений и имени подынтегральной функции. Таким образом, программа будет состоять из головного модуля и двух модулей внешних функций. Головной модуль представляет собой линейный алгоритм, схема которого приведена на рис. 3.1.

Перейдем к составлению схемы внешней функции $F(a, b, m, y)$. Основу алгоритма вычисления интеграла по формуле (2.1) составляет циклический процесс накопления суммы. Перед началом суммирования следует найти шаг h и присвоить начальное значение переменной S, предназначенной для суммы: $S = (y_0 + y_m)/2$. В результате получится схема, приведенная на рис. 3.2.

Внешняя функция $y(x)$ реализует вычисление простого арифметического выражения, поэтому нет нужды приводить ее схему.

Полученные схемы позволяют определить необходимые для решения задачи переменные, присвоить им имена и тип.

Программа составляется в соответствии со схемами, показанными на рис. 3.1 и 3.2, путем последовательной записи блоков в виде операторов и объявлений языка Фортран. Так как имя функции Y встречается в качестве параметра внешней функции F, то оно должно быть объявлено в головном модуле, как $EXTERNAL Y$.

Отладку программы рекомендуется проводить с применением режима "$DEBUG$".

Расчеты по программе в соответствии с заданием проводятся три раза. Для этого желательно заранее подготовить три варианта данных, вводимых с клавиатуры:

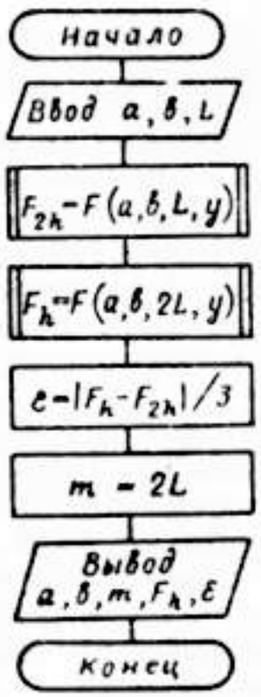

Рис. 3.1

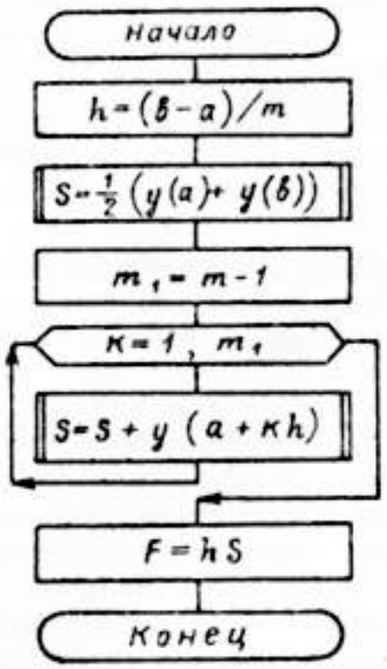

Рис. 3.2

A = I.	B = 2.	L = I
A = I.	B = 2.	L = 5
A = I.	B = 2.	L = I5

Обобщенную квадратурную формуду Симпсона можно представить в следующей удобной для программирования форме:

$$F_h=\frac{h}{3}\left[y_0+y_{2m}+4(y_1+y_3+\dots+y_{2m-1})+2(y_2+y_4+\dots+y_{2m-2})\right], \quad (3.7)$$

где $h=(b-a)/2m$ - шаг интегрирования, $y_\kappa=y(x_\kappa)$ - значения подынтегральной функции в равноотстоящих точках $x_\kappa=a+\kappa h$, $\kappa=0,1,\dots,2m$. Формула Рунге (3.3) в случае обобщенной формулы Симпсона имеет вид

$$\varepsilon_F=\frac{|F_h-F_{2h}|}{15}. \quad (3.8)$$

Рассмотрим применение обобщенной квадратурной формулы Симпсона для решения задачи второго типа.

Пример 2. Вычислить интеграл

$$\mathcal{J}=\int_a^b\frac{e^x}{1+x}dx, \quad a=1, \quad b=2$$

с точностью $\varepsilon=10^{-4}$ по формуле Симпсона. Начальное число разбиений промежутка интегрирования m взять равным 2.

Исходными данными задачи являются величины a, b, ε, m. Их ввод с клавиатуры должен быть предусмотрен в программе. Расчет интеграла производится по формуле (3.7), а расчет погрешности - по формуле (3.8). Как в случае интегрирования по формуле трапеций, программу удобно составить из головного модуля и двух модулей внешних функций: функции $F(a, b, m, y)$, вычисляющей интеграл, и функции $Y(x)$, вычисляющей подынтегральное выражение.

Головной модуль составим в виде цикла последовательных вычислений приближенных значений интеграла $F_h=F(a, b, m, y)$. Для очередного F_h будем проверять условие достижения заданной точности $\varepsilon_F<\varepsilon$, которое с учетом формулы (3.8) можно переписать в виде

$$|F_h-F_{2h}|<15\varepsilon=\varepsilon_1. \quad (3.9)$$

При этом получается следующая последовательность действий:

1) вводятся исходные данные;
2) вычисляется вспомогательное число $\varepsilon_1=15\varepsilon$;

2) по заданному m вычисляется значение интеграла, названное F_{2h};

4) удваивается число дроблений m и вычисляется значение интеграла, названное F_h;

5) проверяется условие (3.9), и если точность еще не достигнута, то только что вычисленное значение интеграла запоминается $F_{2h}=F_h$ (оно понадобится для вычисления погрешности на следующей итерации), и действия повторяются начиная с шага 4;

6) если же заданная точность достигнута, то печатается результат.

Эта последовательность действий в виде схемы представлена на рис. 3.3.

Перейдем к алгоритму внешней функции $F(a, b, m, y)$, вычисляющей интеграл по квадратурной формуле Симпсона (3.7). Эта формула включает в себя две суммы $S_1=y_1+y_3+\ldots+y_{2m-1}$ и $S_2=y_2+y_4+\ldots+y_{2m-2}$, которые содержат разное число слагаемых и входят в формулу (3.7) с разными коэффициентами. Их удобно вычислять с помощью двух различных циклов. Для организации этих циклов вначале следует найти вспомогательные величины $m_1=2m$, $m_2=2m-1$, $m_3=2m-2$ и шаг интегрирования h. В результате получим схему, приведенную на рис. 3.4.

Внешняя функция содержит один оператор присваивания, поэтому для нее схема не нужна.

Предполагается самостоятельно составить программу в соответствии со схемами, приведенными на рис. 3.4, произвести отладку и расчеты.

3. <u>Вычисление интеграла по формулам Гаусса</u>. В отличие от формул трапеций и Симпсона у квадратурных формул Гаусса узлы не являются равноотстоящими. Их выбор подчинен условию достижения наивысшей точности. Формулы Гаусса при различном количестве узлов n имеют общий вид:

$$F=\frac{b-a}{2}\sum_{k=1}^{n}C_k^{(n)}y\left(\frac{a+b}{2}+\frac{b-a}{2}z_k^{(n)}\right). \qquad (3.10)$$

Узлы $z_k^{(n)}$ и коэффициенты $C_k^{(n)}$ приведены в табл. 3.1.

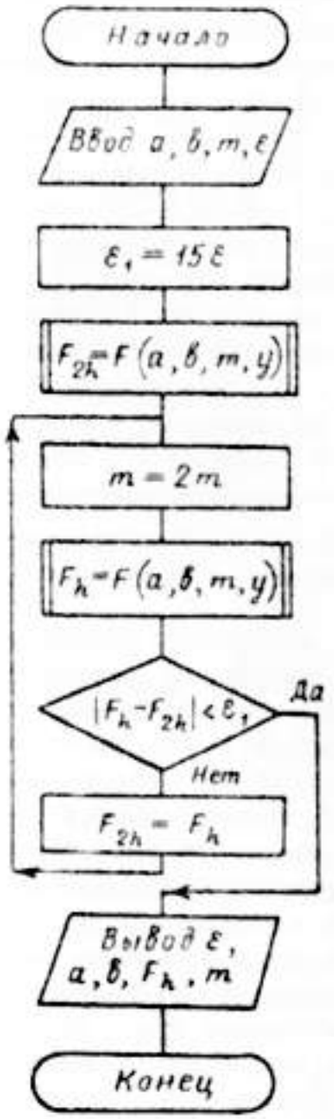

Рис. 3.3

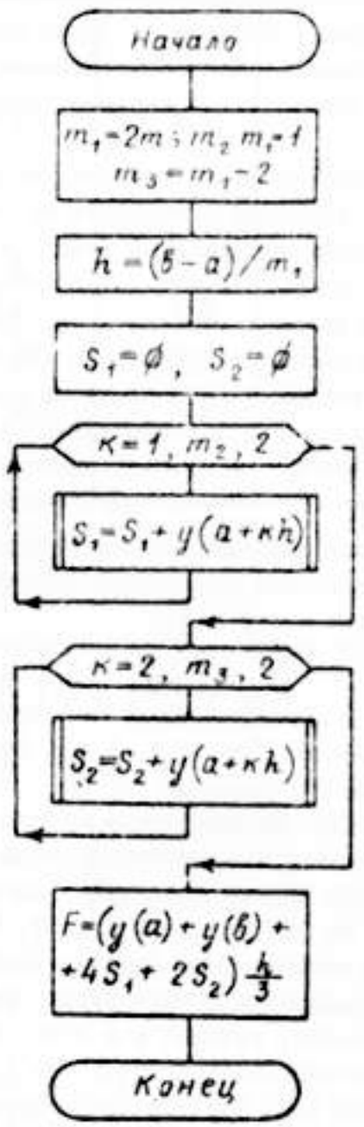

Рис. 3.4

Таблица 3.I

n	k	$z_k^{(n)}$	$C_k^{(n)}$
3	I,3	$\pm$ 0,77459667	$\frac{5}{9}$ = 0,55555556
	2	0	$\frac{8}{9}$ = 0,88888889
4	I,4	$\pm$ 0,86II363I	0,34785484
	2,3	+ 0,33988I04	0,652I45I6
5	I,5	$\pm$ 0,90617985	0,23692688
	2,4	$\pm$ 0,53846931	0,47862867
	3	$\pm$ 0	0,56888889

Каждому значению n соответствует свой набор узлов и коэффициентов, определяющий конкретную формулу Гаусса. Например, в случае $n = 3$, формула имеет вид

$$F = \frac{b-a}{2}\left[\frac{5}{9}\,y\left(\frac{a+b}{2} + \frac{b-a}{2}\cdot 0{,}7746\right) + \frac{8}{9}\,y\left(\frac{a+b}{2}\right) + \frac{5}{9}\,y\left(\frac{a+b}{2} - \frac{b-a}{2}\cdot 0{,}7746\right)\right].$$

Обобщенных формул Гаусса не существует, однако в целях увеличения точности в программах применяют тот же прием, что при выводе обобщенных формул трапеций и Симпсона. Промежуток интегрирования разбивают на m участков длиной $h = (b-a)/m$, к каждому участку применяют квадратурную формулу (3.I0) с $n = 3$, 4 или 5, а затем складывают результаты. Чем больше участков (чем меньше h), тем точнее результат, который и в этом случае можно обозначить F_h. Если у подынтегральной функции $y(x)$ существуют непрерывные производные до $(2n)$-го порядка, то справедлива формула Рунге (3.3) при $r = 2n-1$, которая позволяет приближенно вычислить погрешность результата ε_F:

$$\varepsilon_F = \frac{|F_h - F_{2h}|}{2^{2n-1}-1} \qquad (3.II)$$

Пример 3. Вычислить интеграл $\int_1^2 \frac{e^x}{1+x}\,dx$

по формуле Гаусса с $n = 4$ при двух вариантах разбиения промежутка $m = 2$ и 4. Найти погрешности полученных результатов.

Программу удобно составить из трех модулей: головного модуля, модуля внешней функции, вычисляющей интеграл при определенном числе разбиений промежутка m, и модуля внешней функции, вычисляющей значение $y(x) = e^x/(1+x)$.

В головном модуле программы следует объявить массивы коэффициентов C_k и узлов Z_k и присвоить им конкретные значения, соответствующие формуле Гаусса при $n = 4$. Здесь же предусматривается ввод с клавиатуры пределов интегрирования a, b и числа m. Такая организация ввода данных позволит производить расчеты как для различных пределов интегрирования, так и для различных вариантов разбиения промежутка [a, b]. Вычисления интеграла производятся два раза с разбиением на m участков (-значение F_h) и на $m/2$ участков (- значение F_{2h}). По полученным значениям с помощью формулы (3.II), которая при $n = 4$ примет вид

$$\varepsilon_F = \frac{|F_h - F_{2h}|}{127},$$

вычисляется погрешность. Затем печатаются F_h и ε_F. Схема головного модуля приведена на рис. 3.5.

Внешняя функция, вычисляющая интеграл по формуле Гаусса с n слагаемыми, должна иметь параметрами n, a, b, m, а также имена массивов коэффициентов C и узлов Z и имя внешней функции Y. Алгоритм расчета представляет собой двойной цикл. Внешний цикл обеспечивает перебор всех m участков, на которые разбит промежуток $[a, b]$. Границами каждого такого участка $[a_i, b_i]$ будут $a_i = a + (i-1)h$ и $b_i = a_i + h$ $(i = 1, 2, \ldots m)$. Внутренний цикл позволяет для отдельного участка вычислить интегральную сумму по формуле (3.I0), которая в этом случае принимает вид

$$F_i = \frac{h}{2}\sum_{k=1}^{n} C_k^{(n)}\, y\left(\frac{a_i+b_i}{2} + \frac{h}{2}\,z_k^{(n)}\right).$$

Затем F_i прибавляется к величине S, составляющей полное значение интеграла. Перед циклами необходимо вычислить шаг h и задать начальные значения $S = 0$ и $F_1 = 0$. В результате получится схема, показанная на рис. 3.6.

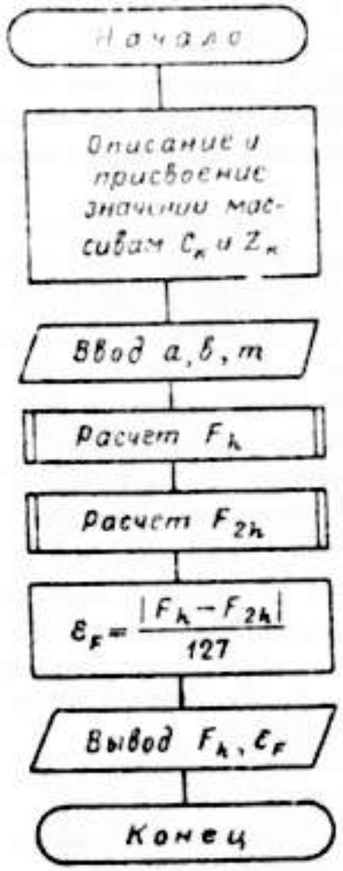

Рис. 3.5

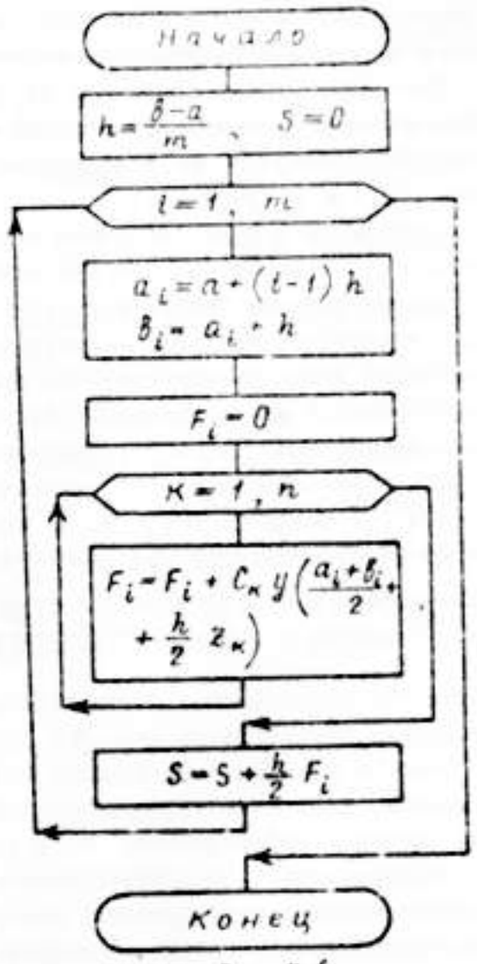

Рис. 3.6

Схемы, приведенные на рис. 3.5 и 3.6, позволяют определить необходимые для решения задачи переменные и массивы, присвоить им имена и тип.

Программа составляется в соответствии со схемами, показанными на рис. 3.5 и 3.6, путем последовательной записи блоков в виде операторов и объявлений на языке Фортран. Так как имя функции Y является параметром внешней функции F, то оно должно быть объявлено в головном модуле, как EXTERNAL Y. Объявление FORMAT для вывода

результатов расчета напишем так, чтобы записи на экране имели вид:

Вычисление интеграла по формуле Гаусса
A = I.0 B = 2.0 ЧИСЛО РАЗБИЕНИЙ M = 4
РЕЗУЛЬТАТ: P = 0.I8327E + 0I ПОГРЕШНОСТЬ = 0.I0E - 05

Отладку программы рекомендуется проводить с применением режима "DEBUG".

Расчеты по программе в соответствии с заданием проводятся два раза. Для этого следует заранее подготовить два варианта исходных данных, вводимых с клавиатуры:

A = I. B = 2. M = 2
A = I. B = 2. M = 4

З А Д А Н И Я

По формуле трапеций с заданным числом дроблений промежутка интегрирования m вычислить следующие интегралы. Найти погрешности интегрирования.

1. $\int_0^1 \cos\frac{\pi x^2}{2}\,dx$, m = I0, 20, 30.
2. $\int_0^1 \frac{\ln(1+x)}{1+x^2}\,dx$; m = I0, 20, 30.
3. $\int_0^1 \frac{arctgx}{x+1}\,dx$; m = 2, I0, 20.
4. $\int_0^1 \frac{dx}{\sqrt{(1+x^2)(1-0{,}25\cos x^2)}}$; m = I0, 20, 30.
5. $\int_0^1 \frac{dx}{\sqrt{(1+x^2)(1-0{,}75x^2)}}$; m = I0, 20, 40.
6. $\int_0^1 \cos\frac{x^2}{4}\,dx$; m = 2, I0, 20.

7. $\int_0^1 \frac{\ln(1+x^2)}{1+x}\,dx$, $m = 2, 10, 20.$

8. $\int_0^1 \frac{(\operatorname{arctg} x)^2}{1+x}\,dx$; $m = 4, 12, 36.$

9. $\int_0^{\pi/3} \frac{dx}{\sqrt{1-0{,}25\sin^2 x}}$; $m = 6, 12, 24.$

10. $\int_0^{1/2} \sqrt{\frac{1-0{,}25x^2}{1-x^2}}\,dx$; $m = 10, 20, 30.$

11. $\int_0^{\pi} \frac{dx}{1+x+\sqrt{1+\sin x}}$; $m = 20, 30, 40.$

12. $\int_0^1 \sqrt{x}\,\sin x\,dx$; $m = 10, 20, 30.$

В следующих задачах вычислить интегралы по формуле Симпсона с заданной абсолютной погрешностью ε и начальным числом дробления промежутка m.

13. $\int_0^{1/2} \sqrt{\frac{1-0{,}75x^2}{1-x^2}}\,dx$; $\varepsilon = 10^{-5}$ $m = 2.$

14. $\int_2^3 \frac{dx}{1+\sqrt{\ln x}}$; $\varepsilon = 10^{-5}$ $m = 2.$

15. $\int_0^1 \sqrt{x}\,\cos dx$; $\varepsilon = 10^{-5}$ $m = 4.$

16. $\int_0^{\pi/2} \frac{dx}{x+\sqrt{1+\cos x}}$; $\varepsilon = 10^{-4}$ $m = 2.$

17. $\int_0^{\pi} \frac{dx}{1+\sin^3 x}$; $\varepsilon = 10^{-4}$ $m = 2.$

18. $\int_0^{\pi/2} \frac{dx}{1+\cos^3 x}$; $\varepsilon = 10^{-5}$ $m = 2.$

19. $\int_0^1 e^{-5x^3+x+0{,}5}\,dx$; $\varepsilon = 10^{-5}$, $m = 4.$

20. $\int_0^1 e^{-4x^3+2x+1}\,dx$; $\varepsilon = 10^{-5}$, $m = 2.$

21. $\int_0^{\pi} \frac{\sin x}{x^2+1}\,dx$; $\varepsilon = 10^{-5}$, $m = 4.$

22. $\int_0^1 \frac{\ln(1+\sqrt{x})}{\sqrt{x+1}}\,dx$; $\varepsilon = 10^{-4}$, $m = 4.$

23. $\int_0^{\pi/2} \sqrt{1-0{,}5\sin^2 x}\,dx$; $\varepsilon = 10^{-5}$, $m = 2.$

24. $\int_0^1 \frac{\sqrt{4+x^2}}{1+\cos^4 x}\,dx$; $\varepsilon = 10^{-5}$, $m = 2.$

25. $\int_0^1 \frac{e^{-x^2}\sin 3x}{3+x^2}\,dx$; $\varepsilon = 10^{-5}$, $m = 4.$

По формуле Гаусса с n узлами для заданного числа разбиений промежутка интегрирования m вычислить следующие интегралы. Найти погрешность интегрирования.

26. $n = 3$, $\int_0^{\pi/2} \frac{dx}{1+\cos^2 x}$; $m = 4, 6.$

27. $n = 3$, $\int_0^1 e^{-5x^3+x+0.5}\,dx$; $m = 4, 6.$

28. $n = 4$, $\int_0^1 \frac{\sin x}{x^2+1}\,dx$; $m = 2, 4.$

29. $n = 4$, $\int_0^1 \frac{\ln(1+\sqrt{x})}{\sqrt{x+1}}\,dx$; $m = 2, 4.$

30. $n = 5$, $\int_0^{\pi/2} \sqrt{1-0{,}5\sin^2 x}\,dx$; $m = 2, 4.$

31. $n = 3$, $\int_2^3 \frac{dx}{1+\sqrt{\ln x}}$; $m = 4, 6.$

32. $n = 4$, $\int_0^1 \frac{dt}{\sqrt{(t^2+1)(3t^2+4)}}$; $m = 2, 4.$

33. $n = 5$, $\int_0^1 \frac{(x^2+1,5)^{5/2}}{x^4+1}dx$, $m = 2, 4.$

В следующих задачах вычислить интегралы с абсолютной погрешностью ε, применив квадратурную формулу Гаусса с n узлами.

34. $\int_0^{\pi} (\sin x)^3\sqrt{4-x^2}\,dx$; $\varepsilon = 10^{-5}$, $n = 3$.

35. $\int_1^2 e^x \ln\left(x + x^2 + \frac{1}{x}\right)dx$; $\varepsilon = 10^{-5}$, $n = 4$.

36. $\int_0^1 \ln(1+\sin t)\,dt$; $\varepsilon = 10^{-5}$, $n = 3$.

37. $\int_0^{\pi/2} \sqrt{3-1,5x}\,\cos 1,5x\,dx$; $\varepsilon = 10^{-5}$, $n = 4$.

38. $\int_{-1}^1 \frac{dx}{x+3\sqrt{x+2}}$; $\varepsilon = 10^{-4}$, $n = 3$.

39. $\int_0^{1/2} \frac{dx}{\sqrt{(1-x^2)(1-0,25x^2)}}$; $\varepsilon = 10^{-4}$, $n = 3$.

40. $\int_0^1 \frac{e^{-1,3x^2}}{1+\sin 1,3x}dx$; $\varepsilon = 10^{-5}$, $n = 3$.

СПИСОК ЛИТЕРАТУРЫ

Основная

I. Дорот В.Л. и др. Элементы вычислительной математики. Численное решение алгебраических и трансцендентных уравнений. - Л.: ЛПИ, 1977, с.54.

2. Калиткин Н.Н. Численные методы. - М.: Наука, 1978, с.512.

3. Демидович Б.П., Марон И.А. Основы вычислительной математики. - М.: Наука, 1970 (и другие годы издания), с.664.

4. Дорот В.Л. и др. Элементы вычислительной математики. Интерполирование. Приближенное вычисление интегралов. - Л.: ЛПИ, 1977, с.60.

5. Субботин Ю.Н., Стечкин. Сплайны в вычислительной математике. - М.: Наука, 1976, с.248.

Дополнительная

I. Траубе Дж. Итерационные методы решения уравнений. - М.: Мир, 1965.

2. Бахвалов Н.С. Численные методы. - М.: Наука, 1973, с. 631.

3. Крылов В.И. Приближенное вычисление интегралов. - М.: Наука, 1967.