Министерство высшего и среднего специального образования БССР
БЕЛОРУССКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра кибернетики и вычислительной техники

ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению контрольных работ по курсу „Вычислительная математика и программирование на ЭВМ"
для студентов специальностей 0502, 0503
заочной формы обучения

Минск 1987

Министерство высшего и среднего специального
образования БССР

БЕЛОРУССКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра кибернетики и вычислительной техники

ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению контрольных работ по курсу
"Вычислительная математика и программирование на ЭВМ"
для студентов специальностей 0502, 0503
заочной формы обучения

Минск 1987

УДК ~~683.3~~

В работе изложены вопросы использования численных методов решения прикладных задач, разработки алгоритмов и написания программ для ЭВМ на алгоритмическом языке программирования Фортран. Приведены варианты контрольных заданий и методические указания по их выполнению.

Изложение сопровождается примерами решения конкретных задач.

Составили:

Ф.М.Бабушкин, О.В.Бугай, В.В.Выговский,
Н.Н.Гурский, А.П.Михалевич

Рецензенты:
А.А.Петровский, С.Н.Фурсенко

1. КОНТРОЛЬНЫЕ РАБОТЫ

По курсу необходимо выполнить три контрольные работы. Первая охватывает численные методы, вопросы разработки алгоритмов и схем алгоритмов решения задач. Вторая работа посвящена программированию на языке Фортран, написанию разветвляющихся и циклических программ, разработке подпрограмм. В третьей работе рассматриваются вопросы использования пакета научных подпрограмм для решения прикладных задач.

Каждый студент выполняет тот вариант задачи, который соответствует последней цифре его шифра.

При решении задач необходимо строго соблюдать требования заданий. Для изучения курса и выполнения контрольных работ рекомендуется пользоваться литературой, список которой приведен ниже. В каждом конкретном задании имеются ссылки на разделы настоящего пособия, содержащие методические указания по способам решения конкретных задач, подробно рассматриваются примеры решения аналогичных задач. Предварительная проработка этих разделов существенно облегчит выполнение задания.

Литература

1. Вычислительная техника в инженерных и экономических расчетах / А.В.Петров, В.Е.Алексеев и др.; Под ред. А.В.Петрова. - М.: Высш.школа, 1984. - 320 с.

2. Алексеев В.Е., Ваулин А.С., Петрова Г.Б. Вычислительная техника в инженерных и экономических расчетах. Сборник задач и упражнений. - М.: Высш.школа, 1984. 136 с.

3. Фурунжиев Р.И. Вычислительная техника и ее применение. - Мн.: Вышэйшая школа, 1984. - 364 с.

4. Фурунжиев Р.И. Вычислительная техника. Практикум. - Мн.: Вышэйшая школа, 1985. - 254 с.

5. Брич З.С. и др. Фортран ЕС ЭВМ. - М.: Статистика, 1978. - 264 с.

6. Бородич Л.И. и др. Справочное пособие по приближенным методам решения задач высшей математики. - Мн.: Вышэйшая школа, 1986. - 190 с.

7. Сборник задач по методам вычислений/ А.И.Азаров, В.А.Басик, И.Н.Мелешко и др.; Под ред. П.И.Монастырного. - Мн.: Изд-во Белорус.ун-та, 1983. - 288 с.

8. Гусак А.А. Элементы методов вычислений. – Мн.: Изд-во Белорус.ун-та, 1982. – 166 с.

9. Копчёнова Н.В., Марон И.А. Вычислительная математика в примерах и задачах. – М.: Наука, 1972. – 368 с.

Контрольная работа № 1

Решить численным методом задачу, условие которой приведено в табл.1. В процессе решения выполнить следующие этапы:

1. Описать алгоритм решения задачи.
2. Начертить схему алгоритма.
3. Получить численное решение задачи. Все вычисления производить с четырьмя десятичными знаками после запятой.

Описание наиболее употребляемых блоков схем алгоритмов, численных методов решения задач и методические рекомендации по их реализации приведены в разделах 2.1–2.3.

Таблица 1

Вариант	Условие задачи
1	2
0	Используя метод последовательного исключения неизвестных (метод Гаусса), решить систему линейных алгебраических уравнений: $\left.\begin{aligned} 2x_1-4x_2-3{,}25x_3+x_4&=4{,}84,\\ 3x_1-3x_2-4{,}3x_3+8x_4&=8{,}89,\\ x_1-5x_2+3{,}3x_3-20x_4&=-14{,}01,\\ 2{,}5x_1-4x_2+2x_3-3x_4&=-20{,}29.\end{aligned}\right\}$
1	По формуле левых прямоугольников вычислить интеграл $\int_a^b f(x)dx=\int_{0{,}6}^{1{,}6}\frac{\lg(x^2+1)}{x}dx$, задаваясь шагом интегрирования h = 0,1. Найти аналитическое решение и определить относительную погрешность вычислений.
2	Найти корень уравнения $f(x)=0{,}25x^3+x-1{,}25=0$ на интервале $[0;\ 2]$, используя метод хорд. Начальным

I	2
	приближением x_0 задаться самостоятельно, процесс вычисления корня закончить при выполнении условия $$\lvert x_n - x_{n-1}\rvert \le 0{,}001.$$
3	По формуле правых прямоугольников вычислить интеграл $$\int_a^b f(x)\,dx = \int_0^2 x^2\sqrt{4-x^2}\,dx,$$ задаваясь шагом интегрирования $h = 0{,}2$. Найти аналитическое решение и определить относительную погрешность вычислений.
4	Найти корень уравнения $$f(x) = 3x - 4\ln x - 5 = 0$$ на интервале [2;4], используя метод Ньютона. Начальным приближением x_0 задаться самостоятельно, процесс вычисления корня закончить при выполнении условия $$\lvert x_n - x_{n-1}\rvert \le 0{,}001.$$
5	По формуле трапеций вычислить интеграл $$\int_a^b f(x)\,dx = \int_3^4 \left(x - 2 + \sin\frac{1}{x}\right)dx,$$ задаваясь шагом интегрирования $h = 0{,}1$. Найти аналитическое решение и определить относительную погрешность вычислений.
6	Используя интерполяционный многочлен Лагранжа найти приближенное значение функции $y = f(x)$, заданной таблично x: 0,43; 0,48; 0,55; 0,62; 0,70; 0,75 y: I,64; I,73; I,88; 2,03; 2,23; 2,36 в точке $x = 0.50$.

I	2
7	Используя метод Эйлера, составить таблицу приближенных значений интеграла дифференциального уравнения $$y' = 3y^{2/3}$$ на отрезке $[0; 1]$ с шагом интегрирования $h = 0{,}1$ и начальным условием $y(0) = 0$.
8	Найти корень уравнения $$f(x) = e^x + \ln x - 10x = 0$$ на интервале $[3; 4]$, используя метод итераций. Начальным приближением x_0 задаться самостоятельно, процесс вычисления корня закончить при выполнении условия $$\lvert x_n - x_{n-1} \rvert \le 0{,}001.$$
9	По формуле Симпсона вычислить интеграл $$\int_a^b f(x)dx = \int_0^1 \frac{dx}{\sqrt{1+3x+2x^2}},$$ задаваясь шагом интегрирования $h = 0{,}1$. Найти аналитическое решение и определить относительную погрешность вычислений.

К о н т р о л ь н а я   р а б о т а № 2

Задание I. Составить на Фортране программу вычисления значений функции, приведенной в табл.2. При разработке программы выполнить следующие этапы:

I. Начертить схему алгоритма решения задачи.

2. Присвоить имена переменным.

3. Выбрать спецификации вводимых исходных данных.

4. Выбрать форму печати на алфавитно-цифровом печатающем устройстве исходных данных и результатов счета.

При этом предусмотреть печать фамилии, и.о.студента и его шифра.

5. Написать текст программы. В конце программы записать исходные данные.

Методические рекомендации по решению задачи содержатся в разделе 2.4 настоящего пособия.

Т а б л и ц а   2

Вариант	Условие задачи	Исходные данные к задаче
I	2	3
0	$y=\begin{cases} a\sum_{i=1}^{n} c_i - b\sqrt{\lvert x\rvert}, & \text{если } a > b \\ \dfrac{\sin x - \ln\lvert a\rvert}{2a-b}, & \text{если } a \le b \end{cases}$ где $x=\sqrt[3]{\lvert ab\rvert}$, $n \le 10$	a=-3,48; в = 5,3; $n=4$; $c_I = -I8,6\cdot IO^I$; $c_2 = I,34\cdot IO^2$; $c_3 = 0,325\cdot IO^3$; $c_4 = 286\cdot IO^{-2}$.
I	$y=\begin{cases} \gamma + 2{,}56, & \text{если } \gamma < 3{,}7\cdot IO^3 \\ \Delta + \rho, & \text{если } \gamma \geqslant 3{,}7\cdot IO^3 \end{cases}$ где $\gamma=\sum_{i=1}^{n}\sqrt{\lvert x_i\rvert}$, $n \le 10$	$\Delta = 3,84$; $\rho = 7,7\cdot IO^3$; $n = 3$; $x_I = 0,3\cdot IO^2$; $x_2 = -I,5\cdot IO^I$; $x_3 = 22,5\cdot IO^{-2}$
2	$y=\begin{cases} ax^3+z, & \text{если } z < 0{,}2 \\ 26{,}5+\cos x, & \text{если } z = 0{,}2 \\ \sqrt{x+25}, & \text{если } z > 0{,}2 \end{cases}$	a = I0,2; x = I,357; $n = 6$; $в_I = 0,I5\cdot IO^{-2}$; $в_2 = -I,I3\cdot IO^{-I}$;

I	2	3
	где $Z = \frac{1}{n}\sum_{i}^{n} b_i$, $n \le 100$	$в_3 = 0,58 \cdot 10^2$; $в_4 = 2,8 \cdot 10^0$; $в_5 = 0,48 \cdot 10^1$; $в_6 = 0,451 \cdot 10^3$.
3	$y = \begin{cases} \frac{arctg\,5x}{b \cdot \cos x + \ln a x}, & \text{если } \lvert Z\rvert < 0,5 \\ 2,546 + Z & \text{если } \lvert Z\rvert \geq 0,5. \end{cases}$ где $Z = \prod_{i=1}^{K} d_i$, $K \le 10$	а = 2,38; в = 1,36,5·10^1; х = 2,5·10^3; к = 4; d_1 = 12,8 d_2 = -15,3 d_3 = -16,5: d_4 = 0,5
4	$y = \begin{cases} \sqrt{\lvert a - b\rvert} + z^2, & \text{если } a < Z \\ \ln\lvert Z - x\rvert, & \text{если } a \geq Z \end{cases}$ где $Z = \frac{a+b}{n!}$	а = 1,38; в = 0,51·10^1; х = -1,345α; n = 10
5.	$y = \begin{cases} \sum_{i=1}^{n} x_i - a, & \text{если } Z < 0,6 \\ 2a & \text{если } Z \geq 0,6 \end{cases}$ где $Z = \frac{1}{a}\sum_{i=1}^{n} x_i$, $n \le 15$	а = 12,5·10^3 n = 4 x_1 = 1,3458; x_2 = -0,8; x_3 = 346,1; x_4 = 1,3
6	$y = \begin{cases} \alpha - x/\ell, & \text{если } x^3 < \alpha^2 \\ \sum_{i=1}^{n} b_i + \alpha, & \text{если } x^3 \geq \alpha^2 \end{cases}$ где $\alpha = \sqrt{\lvert x \cdot \ell\rvert}$, $n \le 10$	х = 0,17·10^2; ℓ = -801, n = 5 $в_1$ = 12,5·10^2 $в_2$ = 0,3·10^3 $в_3$ = -3,7·10^{-1} $в_4$ = -328·10^0 $в_5$ = 1,1·10^4

I	2	3
7	$z=\begin{cases}3{,}56+\dfrac{a-b}{x}, & \text{если } y<x-b,\\ y+\sqrt{\lvert a-b\rvert}, & \text{если } y=x-b,\\ \ln(\sqrt{\lvert x-a\rvert}), & \text{если } y>x-b,\end{cases}$ где $y=\dfrac{a+b}{2{,}8}-x$, $x=\sum_{i=1}^{n}c_i$, $n\le 10$	а = –I8,6; в=0,85·I0^{1}; n = 5 c_I= –0,I25·I0^{3} c_2= I2,8·I0^{0} c_3=I,6·I0^{-I} c_4=–7,5·I0^{2} c_5= 0,36I·I0^{I}
8	$y=\begin{cases}0{,}81+\sum_{i}^{n}x_i, & \text{если } z<a+b,\\ a+b-z, & \text{если } z\geqslant a+b,\end{cases}$ где $z=\dfrac{a+b}{(a^2-b)}$, $n\le 20$	а =–I8,6; в=I,26·I0^{2}; n = 4; x_I= –2,56; x_2= 62,3; x_3= –I3,67; x_4= 5,I
9	$y=\begin{cases}\alpha+\beta+K, & \text{если } \alpha\le\beta,\\ \alpha-\cos\beta, & \text{если } \alpha>\beta,\end{cases}$ где $\beta=\dfrac{c_{i\,max}}{i}$, $(i=1,2,\ldots,n,$ $n\le 10)$	α = –I,2·I0^{2}; к=8; n = 5 c_I= 0,I35; c_2= –I,27; c_3= 8,366; c_4= 0; c_5= 3,5.

Задача 2. Решить задачу, сформулированную в табл.3. При работе над задачей выполнить следующие этапы:

I. Присвоить имя подпрограмме.

2. Начертить схемы алгоритмов задач, решаемых подпрограммой и головной программой.

3. Присвоить имена переменным.

4. Задать численные значения исходных данных и выбрать спецификации по их вводу.

5. Выбрать форму печати результатов выполнения программы на алфавитно-цифровом печатающем устройстве. Предусмотреть при этом печать фамилии, и.о.студента и его шифра.
6. Написать текст подпрограммы и привести форму обращения к ней, дав описание параметров.
7. Написать текст головной программы. В конце программы записать исходные данные.

Методические рекомендации по решению задачи содержатся в разделе 2.5.

Таблица 3

Вариант	Постановка задачи
I	2
0	Оформить вычисление суммы и количества отрицательных элементов вектора (x_i) $(i=\overline{1,n})$ как подпрограмму SUBROUTINE. Используя эту подпрограмму, вычислить сумму и количество отрицательных элементов вектора (a_i) $(i=\overline{1,10})$.
I	Оформить вычисление количества и суммы положительных элементов в нижнем левом треугольнике матрицы $(x_{ij})(i,j=\overline{1,n})$, включая диагональные элементы, как подпрограмму SUBROUTINE. Используя эту подпрограмму, вычислить количество и сумму положительных элементов в нижнем левом треугольнике матрицы (a_{ij}) $(i,j=\overline{1,5})$.
2	Составить подпрограмму SUBROUTINE для формирования из последовательности $(a_i)(i=\overline{1,n})$ новой последовательности $(b_j)(j=\overline{1,n})$, где $b_j=\sum_{k=1}^{j} a_k$, $(j=\overline{1,n})$. Используя эту подпрограмму, сформировать из последовательности $(x_i)(i=\overline{1,n})$ новую последовательность (y_j) $(j=\overline{1,n})$, где $y_j=\sum_{k=1}^{j} x_k$, $(j=\overline{1,n})$.
3	Оформить нахождение координат центра тяжести материальных точек

I	2
	$x_c=\frac{\sum_{i=1}^{n} x_i m_i}{\sum_{i=1}^{n} m_i}$, $y_c=\frac{\sum_{i=1}^{n} y_i m_i}{\sum_{i=1}^{n} m_i}$ как подпрограмму SUBROUTINE. Используя эту подпрограмму, найти координаты центра тяжести трех материальных точек.
4	Оформить нахождение суммы тех строк матрицы (x_{ij}) ($i,j=\overline{1,n}$), у которых на главной диагонали расположены нулевые элементы как подпрограмму SUBROUTINE. Используя эту подпрограмму, найти сумму тех строк матрицы (a_{ij}) ($i,j=\overline{1,5}$), у которых на главной диагонали расположены нулевые элементы.
5	Оформить определение максимального по численному значению элемента последовательности (x_i)($i=\overline{1,n}$) и его порядкового номера как подпрограмму SUBROUTINE. Используя эту подпрограмму, определить максимальный по численному значению элемент последовательности (y_i) ($i=\overline{1,10}$).
6	Оформить определение минимального по численному значению элемента последовательности (x_i) ($i=\overline{1,n}$) и его порядкового номера как подпрограмму SUBROUTINE. Используя эту подпрограмму, определить минимальный по численному значению элемент последовательности (y_i) ($i=\overline{1,20}$) и его порядковый номер.
7	Оформить вычисление сумм столбцов матрицы размера $m \times n$ как подпрограмму SUBROUTINE. Используя эту подпрограмму, вычислить суммы столбцов матрицы 5x3.
8	Оформить подсчет количества нулевых элементов в каждой строке матрицы размером $m \times n$ как подпрограмму SUBROUTINE. Используя эту подпрограмму, вычислить количество нулевых элементов в каждой строке матрицы размером 4x5.
9	Оформить вычисление сумм элементов четных строк матрицы размера $m \times n$ как подпрограмму SUBROUTINE. Используя ее, решить такую задачу для матрицы 5 x 3

Контрольная работа № 3

Составить на Фортране с использованием пакета научных подпрограмм программу решения задачи, условие которой приведено в табл.4. При разработке программы выполнить следующие этапы:

1. Начертить схему алгоритма решения задачи. Для реализации численного метода решения использовать указанную в условии подпрограмму.
2. Присвоить имена переменным.
3. Выбрать спецификации вводимых исходных данных.
4. Выбрать форму печати результатов выполнения программы на алфавитно-цифровом печатающем устройстве. Предусмотреть при этом печать фамилии, и.о. студента и его шифра.
5. Написать текст программы. В конце программы записать исходные данные.

Методические рекомендации по использованию пакета научных подпрограмм в программах решения прикладных задач содержатся в разделе 2.6.

Таблица 4

Вариант	Условие задачи
1	2
0	Решить методом Гаусса (подпрограмма *SIMQ*) систему линейных алгебраических уравнений: $\left.\begin{array}{l} x_1+x_2+x_3+x_4 = 10 \\ 2x_1-x_2+2x_3-x_4 = 2 \\ 3x_1+2x_2+x_3-3x_4 = 2 \\ 4x_1+2x_2-5x_3+2x_4 = 1 \end{array}\right\}$ Решить с точностью $\varepsilon = 10^{-3}$ на интервале $[0;6]$ методом деления пополам (подпрограмма *RTIL1*) нелинейное уравнение $f(x)= \ln(0{,}9146x)-1{,}038x+2{,}5$ Максимальное количество заданных итераций – 20. Для вычисления значений функции $f(x)$ разработать внешнюю подпрограмму типа *FUNCTION* с именем *FCT* .

1	2

Вычислить значение определенного интеграла

$$Z=\int_a^b f(x)\,dx=\int_0^1 \frac{x\,\mathrm{arctg}\,x}{\sqrt{1+x^2}}\,dx$$

методом Симпсона (подпрограмма QSF) с шагом h = 0,I. В программе предусмотреть формирование массива входных значений функции

$$y_i=f(x_i)=\frac{x_i\,\mathrm{arctg}\,x_i}{\sqrt{1+x_i^2}}\ ,\quad i=1,\ldots,n.$$

Здесь $n=\frac{b-a}{h}+1,\ \ x_1=a=0,\ x_n=b=1.$

Решить с точностью ε = 10^{-3} на интервале [-2;2] методом деления пополам (подпрограмма $RTILI$) нелинейное уравнение

I $f(x)$= $1{,}2755x^3-3{,}601x^2-1{,}37x+6{,}76$.

Максимальное количество заданных итераций – 20. Для вычисления значений функции $f(x)$ разработать внешнюю подпрограмму типа $FUNCTION$ с именем FCT .

Провести интерполирование функции $y=f(x)$ методом Лагранжа (подпрограмма ALI). Функция представлена таблицей экспериментальных значений:

x	0	I5	30	45	60	75	90	I05
$f(x)$	800	718	665	62I	586	556	532	5I0

Интерполирование провести для значения аргумента x=68. Верхняя граница абсолютной ошибки ε = 10^{-4}.

Решить на интервале [I; 20] методом Рунге-Кутта (подпрограмма $RK2$) дифференциальное уравнение

2

$$y'=f(x,y)=\sin x-y/x$$

с начальными условиями x_0 = I, y_0 = I и шагом интегрирования h= 0,I. Количество шагов между двумя сохраняемыми в памяти величинами К = I0. Для вычисления значений функции $f(x,y)$

I	2

разработать внешнюю подпрограмму типа *FUNCTION* с именем *FONC*.

Решить с точностью $\varepsilon = 10^{-4}$ методом последовательных приближений (подпрограмма *RTWI*) нелинейное уравнение

$$x = \varphi(x) = \frac{1}{3 + \sin 3{,}6x} .$$

Начальное приближение корня $x_0 = 0{,}51$. Максимальное количество заданных итераций – 20. Для вычисления значений функции $\varphi(x)$ разработать внешнюю подпрограмму типа *FUNCTION* с именем *FUNC*.

3 Найти значение определенного интеграла

$$Z = \int_a^b f(x)\,dx = \int_{\frac{\pi}{6}}^{\frac{\pi}{3}} (tg^2 x + ctg^2 x)\,dx$$

методом трапеций (подпрограмма *QTFE*) с шагом $h = \frac{\pi}{36}$. В программе предусмотреть формирование массива входных значений функции

$$y_i = f(x_i) = tg^2 x_i + ctg^2 x_i,\ i = 1, \ldots, n .$$

Здесь $n = \frac{b-a}{h} + 1,\ x_1 = a = \frac{\pi}{6},\ x_n = b = \frac{\pi}{3},$
$\pi = 3{,}1415927$.

Провести интерполирование функции $y = f(x)$ методом Ньютона (подпрограмма *LINTX*). Функция представлена таблицей экспериментальных значений:

x	-2	3	4,5	I2	I5	I8,5	20
$f(x)$	-I7	-3	I,2	I,8	3	4,5	7

Интерполирование провести для значения аргумента $x = 9{,}8$.

4 Решить на интервале [3; I0] методом Рунге-Кутта (подпрограмма *R*K2) дифференциальное уравнение

$$y' = f(x,y) = x e^{-x^2} - 2xy$$

с начальными условиями $x_0 = 1$, $y_0 = 2{,}23$ и шагом интегрирования $h = 0{,}1$. Количество шагов между двумя сохраняемыми в памяти величинами К=5. Для вычисления значений функции $f(x,y)$

1	2

разработать внешнюю подпрограмму типа *FUNCTION* с именем *F* .

Вычислить значение определенного интеграла

$$Z = \int_a^b f(x)\,dx = \int_0^2 x^2 \ln(x^2+1)\,dx$$

методом Симпсона (подпрограмма *QSF*) с шагом h =0,I. В программе предусмотреть формирование массива входных значений функции

$$y_i = f(x_i) = x_i^2 \ln(x_i^2+1),\ i=1,\ldots,n.$$

Здесь $n = \frac{b-a}{h}+1,\ x_1 = a = 0,\ x_n = b = 2.$

Решить с точностью $\varepsilon = 10^{-3}$ методом последовательных приближений (подпрограмма *RTWI*) нелинейное

5 уравнение

$$x = \varphi(x) = 0{,}2\,(x^2+4).$$

Начальное приближение корня x_0 = 0,8. Максимальное количество заданных итераций -30. Для вычисления значений функции $\varphi(x)$ разработать внешнюю подпрограмму типа *FUNCTION* с именем *FI* .

Провести интерполирование функции $y=f(x)$ методом Лагранжа (подпрограмма *ALI*). Функция представлена таблицей экспериментальных значений:

x	-2,3	0	I,I	4,8	7,3	9,2	II,4
$f(x)$	-I2,5	8,6	I3,4	I5,I	2I,4	24,2	28,3

6 Интерполирование провести для значения аргумента x=I,5. Верхняя граница абсолютной ошибки $\varepsilon = 10^{-4}$.

Решить на интервале [0; 5] методом Рунге-Кутта (подпрограмма *RK2*) дифференциальное уравнение

$$y' = f(x,y)\sin 2x - y\,\mathrm{tg}\,x$$

с начальными условиями x_0 = 0, y_0 = -I и шагом интегри-

I	2
	рования h = 0,I. Количество шагов между двумя сохраняемыми в памяти величинами K=5. Для вычисления значений функции $f(x,y)$ разработать внешнюю подпрограмму типа *FUNCTION* с именем *F* .
	Решить на интервале [0;I0] методом Рунге-Кутта (подпрограмма *RK2*) дифференциальное уравнение $$y' = f(x,y) = \cos x - \sin x - y$$ с начальными условиями x_0=0, y_0=2 и шагом интегрирования h =0,I. Количество шагов между двумя сохраняемыми в памяти величинами K=I0. Для вычисления значений функции $f(x,y)$ разработать внешнюю подпрограмму типа *FUNCTION* с именем *FUNC* .
7	Провести интерполирование функции $y=f(x)$ методом Ньютона (подпрограмма *LINTX*). Функция представлена таблицей экспериментальных значений:

x	2,34	5,I6	7.03	8,42	9,6I	I0,I2	II,35	I2,I2
$f(x)$	I5,I6	25,03	32,I8	37,II	44,82	5I,62	50,I3	73,I6

Интерполирование провести для значения аргумента x= 5,5.

Решить с точностью ε = 10^{-3} методом последовательных приближений (подпрограмма *RTWI*) нелинейное уравнение

$$x = \varphi(x) = 2 - \sin 1/x .$$

Начальное приближение корня x_0 = I,3077. Максимальное количество заданных итераций - 20. Для вычисления значений функции $\varphi(x)$ разработать внешнюю подпрограмму типа *FUNCTION* с именем *F* .

8 Решить методом Гаусса (подпрограмма *SIMQ*) систему линейных алгебраических уравнений

$$\left.\begin{aligned} 1{,}84x_1 + 2{,}25x_2 + 2{,}53x_3 &= -6{,}09 \\ 2{,}32x_1 + 2{,}60x_2 + 2{,}82x_3 &= -6{,}98 \end{aligned}\right\}$$

I	2
	$1{,}83x_1 + 2{,}06x_2 + 2{,}24x_3 = -5{,}52$
9	Вычислить значение определенного интеграла $$z = \int_a^b f(x)dx = \int_1^2 \frac{\sqrt{x^2-0{,}16}}{x}dx$$ методом трапеций (подпрограмма *QTFE*) с шагом h =0,1. В программе предусмотреть формирование массива входных значений функции $$y_i = f(x_i) = \frac{\sqrt{x_i^2-0{,}16}}{x_i}, \quad i = 1, \dots, n.$$ Здесь $n = \frac{b-a}{h} + 1, \quad x_1 = a = 1, \; x_n = b = 2.$ Решить с точностью ε = 10^{-3} на интервале [0; 1] методом деления пополам (подпрограмма *RTILI*) нелинейное уравнение $$f(x) = \sqrt{1-x} - tg\,x = 0.$$ Максимальное количество заданных итераций – 50. Для вычисления значений функции $f(x)$ разработать внешнюю подпрограмму типа *FUNCTION* с именем *FCT* .

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КОНТРОЛЬНЫХ РАБОТ

2.I. Численные методы

Подробное изложение методов решения различных задач содержится в учебных пособиях [6] - [9] . В данном разделе приведены вычислительные схемы, реализующие методы решения задач контрольной работы № I.

2.I.I. Решение систем линейных алгебраических уравнений

Одним из распространенных методов решения систем уравнений является метод последовательного исключения неизвестных (метод Гаусса). Вычислительная схема метода для системы четырех уравнений с четырьмя неизвестными

$$\left.\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= a_{15} \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= a_{25} \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= a_{35} \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= a_{45} \end{aligned}\right\}$$

приведена в табл.5,6.

Т а б л и ц а 5

Коэффициенты при неизвестных				Свободные члены	Контрольные суммы
x_1	x_2	x_3	x_4		
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	c_1
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	c_2
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	c_3
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	c_4
I	α_{12}	α_{13}	α_{14}	α_{15}	β_1
	a'_{22}	a'_{23}	a'_{24}	a'_{25}	c'_2
	a'_{32}	a'_{33}	a'_{34}	a'_{35}	c'_3
	a'_{42}	a'_{43}	a'_{44}	a'_{45}	c'_4
	I	α_{23}	α_{24}	α_{25}	β_2
		a''_{33}	a''_{34}	a''_{35}	c''_3
		a''_{43}	a''_{44}	a''_{45}	c''_4

Продолжение табл.5

I	2	3	4	5	6
		I	α_{34}	α_{35}	β_3
			a'''_{44}	a'''_{45}	c'''_4
			I	α_{45}	β_4
			I	x_4	$\bar{x}_4$
		I		x_3	$\bar{x}_3$
	I			x_2	$\bar{x}_2$
I				x_I	$\bar{x}_I$

Расчетные соотношения для получения элементов табл.5 содержатся в табл.6.

Таблица 6

Вычислительные формулы		Контрольные соотношения
$c_i = \sum_{j=1}^{5} a_{ij}$	$(i = I,2,3,4)$	
$\alpha_{1j} = \dfrac{a_{1j}}{a_{11}}$; $\beta_1 = \dfrac{c_1}{a_{11}}$	$(j = 2,3,4,5)$	$1+\alpha_{12}+\alpha_{13}+\alpha_{14}+\alpha_{15}=\beta_1$
$a'_{ij} = a_{ij} - a_{i1}\alpha_{1j}$; $c'_i = c_i - a_{i1}\beta_1$	$(i = 2,3,4)$ $(j = 2,3,4,5)$	$a'_{i2}+a'_{i3}+a'_{i4}+a'_{i5}=c'_i$ $(i = 2,3,4)$
$\alpha_{2j} = \dfrac{a'_{2j}}{a'_{22}}$; $\beta_2 = \dfrac{c'_2}{a'_{22}}$	$(j = 3,4,5)$	$1+\alpha_{23}+\alpha_{24}+\alpha_{25}=\beta_2$
$a''_{ij} = a'_{ij} - a'_{i2}\alpha_{2j}$; $c''_i = c'_i - a'_{i2}\beta$	$(i = 3,4)$ $(j = 3,4,5)$	$a''_{i3}+a''_{i4}+a''_{i5}=c''_i$ $(i = 3,4)$
$\alpha_{3j} = \dfrac{a''_{3j}}{a''_{33}}$	$(j = 4,5)$	$1+\alpha_{34}+\alpha_{35}=\beta_3$

I	2
$\beta_3 = \dfrac{c_3'''}{a_{33}''}$	
$a_{4j}''' = a_{4j}'' - a_{43}''\alpha_{3j}$ $(j = 4,5)$ $c_4''' = c_4'' - a_{43}''\beta$	$a_{44}''' + a_{45}''' = c_4'''$
$\alpha_{45} = \dfrac{a_{45}'''}{a_{44}'''}$ $\beta_4 = \dfrac{c_4'''}{a_{44}'''}$	$1 + \alpha_{45} = \beta_4$
$x_4 = \alpha_{45}$ $x_3 = \alpha_{35} - \alpha_{34}x_4$ $x_2 = \alpha_{25} - \alpha_{24}x_4 - \alpha_{23}x_3$ $x_1 = \alpha_{15} - \alpha_{14}x_4 - \alpha_{13}x_3 - \alpha_{n}x_2$ $\bar{x}_4 = \beta_4$ $\bar{x}_3 = \beta_3 - \alpha_{34}\bar{x}_4$ $\bar{x}_2 = \beta_2 - \alpha_{24}\bar{x}_4 - \alpha_{23}\bar{x}_3$ $\bar{x}_1 = \beta_1 - \alpha_{14}\bar{x}_4 - \alpha_{13}\bar{x}_3 - \alpha_{12}\bar{x}_2$	$I + x_4 = \bar{x}_4$ $I + x_3 = \bar{x}_3$ $I + x_2 = \bar{x}_2$ $I + x_I = \bar{x}_I$

2.I.2. Решение нелинейных уравнений

Вычисление корней (решений) алгебраических и трансцендентных нелинейных уравнений вида

$$f(x) = 0$$

с заданной наперед точностью обычно осуществляют путем применения какой-либо итерационной процедуры, состоящей в построении числовых последовательностей x_n, сходящихся к искомому корню x^* уравнения. Ниже приводятся формулы итерационных процедур различных методов отыскания корней уравнения $f(x) = 0$ на отрезке [a,в].

I. Метод хорд.

Метод хорд требует исследования изменения знака второй производной $f''(x)$ на отрезке [a,в] и принятия соответствующего решения о выборе начального приближения $x=x_0$ и рекуррентной формулы.

Если $f(b)f''(x)>0$ на [а,в], то x_0 = а и

$$x_{n+1}=x_n-\frac{f(x_n)}{f(b)-f(x_n)}(b-x_n).$$

Если $f(a)f''(x)>0$ на [а,в], то x_0 = в и

$$x_{n+1}=a-\frac{f(a)}{f(x_n)-f(a)}(x_n-a).$$

2. Метод Ньютона (метод касательных)

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)},$$

причем $x_0 = \begin{cases} \text{а, если } f(a)f''(x)>0 \text{ на [а,в];} \\ \text{в, если } f(b)f''(x)>0 \text{ на [а,в].} \end{cases}$

Метод Ньютона эффективен в случае, если значение $|f'(x)|$ близ корня x^* достаточно велико, т.е. имеет место большая крутизна функции $y=f(x)$ в окрестности корня.

3. Метод итераций.

Уравнение $f(x)=0$ приводится к виду $x=\varphi(x)$, либо непосредственным выделением x из уравнения $f(x)=0$, либо с помощью преобразования

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

Величину k следует выбирать таким образом, чтобы

$$|k|>\frac{Q}{2},$$

$Q=max|f'(x)|$ на отрезке [а,в].

Итерационный процесс определения корня уравнения $\varphi(x)$ сходится при выполнении условия

$|\varphi'(x)|<1$ на [а,в].

Рекуррентная формула для этого метода имеет вид

$$x_{n+1}=\varphi(x_n), \quad n=0,1,2,\ldots$$

Начальное приближение x_0 задается из промежутка [а,в].

2.1.3. Интерполирование функций

В общем случае задача отыскания интерполирующего многочлена решается таким образом. Пусть функция $f(x)$ определена на отрезке [а;в] и известны значения функции в точках отрезка $x_0, x_1, x_2\ldots, x_n$. Значения функции обозначим $y_0, y_1, y_2\ldots, y_n$; т.е.

$y_0 = f(x_0)$, $y_1 = f(x_1)$, ..., $y_n = f(x_n)$. Точки x_0, x_1, ... , x_n называются узлами интерполяции. Ставится задание найти такой многочлен, значения которого в узлах интерполяции совпадают с соответствующими значениями функции.

Интерполяционная формула Лагранжа

$$L(x) = \sum_{k=0}^{n} y_k \frac{(x-x_0)(x-x_1)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)(x_k-x_1)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)}$$

позволяет найти интерполяционный многочлен n-ой степени.

При вычислении коэффициентов Лагранжа разности удобно располагать следующим образом:

$\boxed{x-x_0}$	x_0-x_1	x_0-x_2	- - -	x_0-x_n
x_1-x_0	$\boxed{x-x_1}$	x_1-x_2	- - -	x_1-x_n
x_2-x_0	x_2-x_1	$\boxed{x-x_2}$	- - -	x_2-x_n
- - -	- - -	- - -	- - -	- - -
x_n-x_0	x_n-x_1	x_n-x_2	- - -	$\boxed{x-x_n}$

2.1.4. Формулы приближенного вычисления интегралов

Пусть требуется вычислить интеграл $\int_a^b f(x)\,dx$, где $f(x)$ – данная функция, непрерывная на отрезке $[a,b]$. Разделим отрезок $[a,b]$ на n равных частей, т.е. n равных частичных отрезков. Длина каждого отрезка (шаг интегрирования) $h = \frac{b-a}{n}$. Точки деления будут $x_0 = a$, $x_1 = a+h$, $x_2 = a+2h$, ..., $x_{n-1} = a+(n-1)h$, $x_n = b$. Эти числа называют узлами. Значения функции в узлах обозначим $y_0, y_1, \ldots, y_n$, т.е. $y_0 = f(a), y_1 = f(x_1), \ldots, y_n = f(b)$. Для оценки погрешности вводится параметр $R_n(f)$ остаточный член формул интегрирования, который определяет разность между точным значением интеграла и его приближенным значением.

1. Формула левых прямоугольников

$$\int_a^b f(x)\,dx \approx h \sum_{i=0}^{n-1} f(x_i),$$

где $x_i = a + ih \quad (i = 0, 1, 2, \ldots, n)$.

2. Формула правых прямоугольников

$$\int_a^b f(x)\,dx \approx h \sum_{i=1}^{n} f(x_i),$$

где $x_i = a + ih \quad (i = 0, 1, 2, \ldots, n)$

3. Формула средних прямоугольников

$$\int_a^b f(x)dx \approx h\sum_{i=0}^{n-1} f\left(x_i + \frac{h}{a}\right),$$

где

$$x_i = a + ih \quad (i = 0, 1, 2, \ldots, n).$$

4. Формула трапеций

$$\int_a^b f(x)dx \approx h\left(\frac{y_0+y_n}{2} + \sum_{i=1}^{n-1} y_i\right),$$

где

$$y_i = f(x_i) \ (i = 0, 1, 2, \ldots, n), \quad x_i = a + ih.$$

5. Формула Симпсона (число n – обязательно четное)

$$\int_a^b f(x)dx \approx \frac{h}{3}\left[y_0 + y_n + 2(y_2 + y_4 + \ldots + y_{n-2}) + 4(y_1 + y_3 + \ldots + y_{n-1})\right],$$

где $x_i = a + ih$, $(i = 0, 1, 2, \ldots, n)$, $y_i = f(x_i)$.

Из всех приведенных формул наиболее точной является формула Симпсона. Так, для получения точности в 2–3 значащие цифры при применении формулы Симпсона достаточно 4–6 ординат, при применении формулы трапеции – 8–12 ординат, при применении формулы прямоугольников – до 24 ординат.

2.1.5. Приближенные методы решения обыкновенных дифференциальных уравнений

Обыкновенным дифференциальным уравнением называется выражение

$$F(x, y, y', y'', \ldots, y^{(n-1)}, y^{(n)}) = 0,$$

где x – независимая переменная, y – искомая функция (от x), $y^{(k)}$ – производные k-го порядка.

Уравнение называется разрешенным относительно старшей производной, если оно имеет вид

$$y^{(n)} = f(x, y, y', y'', \ldots, y^{(n-1)}).$$

Задача отыскания решения уравнения этого вида $y = f(x)$ при начальных условиях $y_0 = y(x_0), y_0' = y'(x_0), \ldots, y_0^{(n-1)} = y^{(n-1)}(x_0)$ называется задачей Коши для обыкновенного дифференциального уравнения

Метод Эйлера (для дифференциальных уравнений первого порядка)

Пусть дано уравнение $y' = f(x, y)$ с начальным условием $y(x_0) = y_0$.

Тогда решение на отрезке $[a, b]$ будем искать по рекуррентной формуле

$$y_{k+1} = y_k + h f(x_k, y_k) \quad (k = 0, 1, 2, \ldots, n-1),$$

где

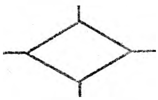
$$y_k = y(x_k), \quad x_k = x_0 + k \cdot h, \quad h = (b-a)/n.$$

2.2. Изображение схем алгоритмов

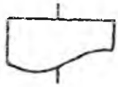
Алгоритм – это последовательность предписаний (правил), однозначно определяющих процесс преобразования исходных данных в результат решения задачи. Основой для разработки алгоритма решения задачи является ее математическая формулировка.

Наиболее наглядный способ представления алгоритмов – их изображение в виде схем – последовательности блоков, предписывающих выполнение определенных функций, и связей между ними. Внутри блоков указывается поясняющая информация, характеризующая выполняемые ими действия. Блоки схемы обычно имеют сквозную нумерацию. Конфигурацию и размер блоков, а также порядок построения схем определяют ГОСТы 19002-80 и 19003-80. В табл.7 приведены некоторые наиболее часто употребляемые блоки и даны пояснения к ним.

Таблица 7

Название символа	Символ	Примечание
1	2	3
Процесс		Соответствует одному или нескольким операторам присваивания
Решение		Проверка условий; соответствует оператору *IF*
Модификация		Начало цикла; соответствует оператору *DO*
Предопределенный процесс		Вычисление по подпрограмме
Перфокарта		Ввод данных с перфокарт или вывод данных на перфокарты; соответствует оператору *READ*

Продолжение табл. 7

I	2	3
		при вводе или при выводе – *WRITE*
Документ		Вывод данных, печать результатов; соответствует оператору *WRITE*
Пуск, останов		Начало, конец, останов, вход и выход в подпрограммах. Этим блоком всегда начинается и заканчивается схема алгоритма

Блоки на схеме алгоритма соединяются линиями связи, указывающими очередность выполнения предписаний алгоритма. Если эти линии направлены сверху вниз или слева направо, то стрелки на них могут не проставляться; при обратном направлении линий связи, т.е. снизу вверх и справа налево, стрелки проставляются обязательно.

Большинство блоков схемы алгоритма имеют один вход (верхняя линия) и один выход (нижняя линия). Блок "решение" имеет два или более выходов (нижняя и боковые линии) в зависимости от количества проверяемых условий. Назначение линий связи блока "модификация" следующее: I – вход в цикл, начало цикла; 2 – нормальный выход из цикла, когда цикл выполняется количество раз, определенное оператором *DO* ; 3 – переход к операторам внутри цикла; 4 – возрат к началу цикла после каждого очередного выполнения операторов цикла.

2.3. Пример решения задачи численным методом

Процедуру выполнения контрольной работы I поясним на следующем примере.

По формуле средних прямоугольников вычислить интеграл

$$\int_a^b f(x)\,dx = \int_2^3 \frac{x^2}{x^3+1}\,dx ,$$

задаваясь шагом интегрирования $h = 0{,}2$. Вычисления проводить с четырьмя десятичными значениями.

I. Описание алгоритма решения задачи.

Воспользуемся описанием алгоритма, приведенного в разделе 2.I.4

$$I_4 = \int_a^b f(x)dx \approx h \sum_{i=0}^{n-1} f\left(x_i + \frac{h}{a}\right),$$

где $\quad n = \dfrac{b-a}{h}$,

$$x_i = a + ih \quad (i = 0, 1, 2, \ldots, n).$$

Введем новое обозначение

$$x_{1i} = x_i + \frac{h}{a} \quad (i = 0, 1, 2, \ldots, n).$$

Тогда формула средних прямоугольников запишется в виде

$$I_4 \approx h \sum_{i=0}^{n-1} f(x_{1i}).$$

2. Составление схемы алгоритма решения задачи.

Приведем формулы разработанного ранее алгоритма

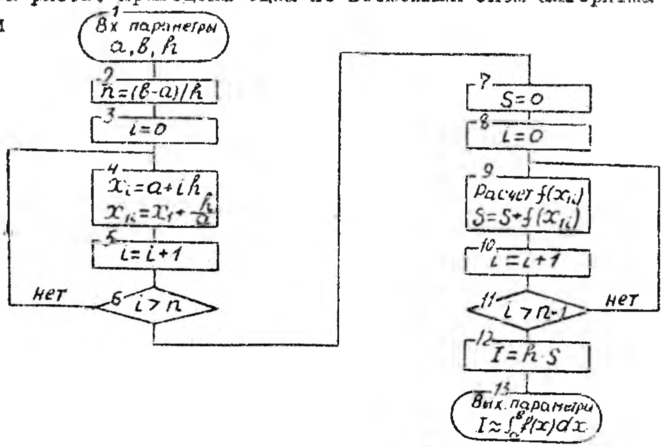
$$n = \frac{b-a}{h}$$

$$x_i = a + ih \quad (i = 0, 1, 2, \ldots, n)$$

$$x_{1i} = x_i + \frac{h}{a} \quad (i = 0, 1, 2, \ldots, n)$$

$$I = h \sum_{i=0}^{n-1} f(x_{1i})$$

На рис.I. приведена одна из возможных схем алгоритма решения задачи



В схеме принято обозначение $\quad S = \sum_{i=0}^{n-1} f(x_{1i})$.

Тогда $\quad I = h \cdot S$.

3. Численное решение задачи

Определим количество отрезков интегрирования

$$n = \frac{b-a}{h} = \frac{3-2}{0,2} = 5$$

Определим x_i по формуле

$$x_i = a + ih = 2 + 0,2i$$

Определим по формуле x_{ii}

$$x_{ii} = x_i + h/a = x_i + \frac{0,2}{2} = x_i + 0,1$$

Результаты вычислений $x_i, x_{ii}, f(x_{ii})$ приведены в табл.8.

Т а б л и ц а 8

i	$x_i = a + ih$	$x_{ii} = x_i + h/a$	x_{ii}^2	x_{ii}^3	$x_{ii}^3 + 1$	$f(x_{ii}) = \frac{x_{ii}^2}{x_{ii}^3+1}$
0	2	2,I	4,4I	9,26I	I0,26I	0,4298
I	2,2	2,3	5,29	I2,I67	I3,I67	0,4018
2	2,4	2,5	6,25	I5,625	I6,625	0,3759
3	2,6	2,7	7,29	I9,683	20,683	0,3524
4	2,8	2,9	8,4I	24,389	25,389	0,3309
5	3,0	3,I	9,6I	29,79I	30,79I	0,312I

Найдем значение интеграла

$$S = \sum_{i=0}^{n-1} f(x_{ii}) = \sum_{i=0}^{4} f(x_{ii}) = (0,4298 + 0,4018 + 0,3759 + 0,3524 + 0,3309) = 1,8908$$

$$I_4 = hS = 0,2 \cdot 1,8908 = 0,3782.$$

Найдем аналитическое значение для интеграла. Выполним замену переменной

$$t = x^4, \quad \text{тогда} \quad dt = 4x^3 dx \quad и$$

$$I_A = \int_2^3 \frac{x^3}{x^4+1} dx = \frac{1}{4} \int_{a_1}^{b_1} \frac{dt}{t+1} .$$

Здесь a_1, b_1 – новые пределы

$$x = 2 \quad t = 2^4 = 16, \quad x = 3 \quad t = 3^4 = 81.$$

Следовательно,

$$I_A = \int_2^3 \frac{x^3}{x^4+1} dx = \frac{1}{4} \int_{16}^{81} \frac{dt}{t+1} = \frac{1}{4} \ln|t+1| \Big|_{16}^{81} = \frac{1}{4} \ln \frac{82}{17} = \frac{1}{4} \ln 4,8235 = 0,25 \cdot 1,5736 = 0,3934.$$

Оценим погрешность применения формулы средних прямоугольников

$$\Delta I = \frac{I_A - I_ч}{I_A} = \frac{0,3934 - 0,3782}{0,3934} = \frac{0,0152}{0,3934} = 0,0386$$

, т.е. погрешность составила 3,86%.

2.4. Программирование разветвляющихся и циклических вычислительных процессов

Методику программирования названных процессов на языке ФОРТРАН рассмотрим на конкретном примере.

Постановка задачи

Составить программу по вычислению *)

$$y = \begin{cases} \sum_{i=1}^{n} a_i + b & при\ \ z < 0,3\,b, \\ \prod_{i=1}^{n} a_i + x & при\ \ z \geqslant 0,3\,b, \end{cases}$$

где $z = \frac{\sqrt{|b - x|}}{n}$, предусмотрев расчет при

$b = -13,62$ $\quad$ $x = 0,3 \cdot 10^2$ $\quad$ $n = 5$

$a_1 = -12,6$ $\quad$ $a_3 = -1,83$ $\quad$ $a_5 = 175,1$

$a_2 = 0,2461$ $\quad$ $a_4 = 0,3$

Решение

1. Введем промежуточные переменные $s = \sum_{i=1}^{n} a_i$ и $p = \prod_{i=1}^{n} a_i$. Тогда исходная формула перепишется в виде

$$y = \begin{cases} s + b & при\ \ z < 0,3\,b, \\ p + x & при\ \ z \geqslant 0,3\,b. \end{cases}$$

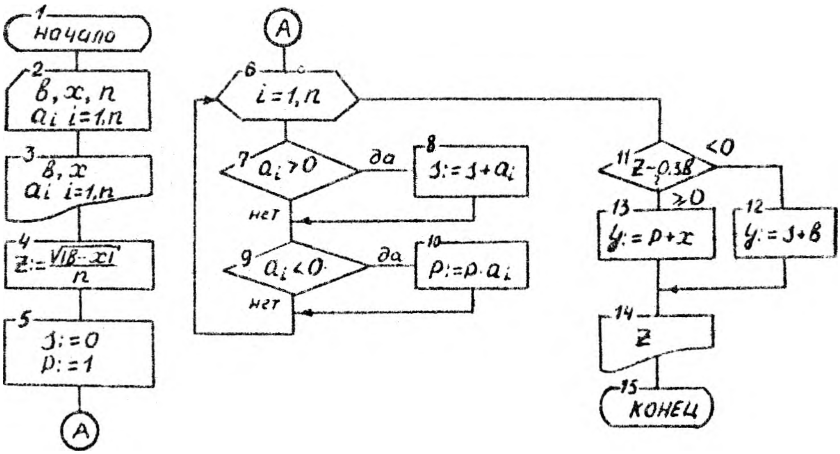
2. В программе предусмотрим:

- ввод исходных данных и вывод их для контроля на печать;
- вычисление z, s, p и y;
- вывод результата счета на печать.

В соответствии с этим алгоритм решения задачи может быть представлен в виде следующей схемы:

---

*) В суммировании участвуют положительные, а в перемножении – отрицательные элементы массива.



3. Присваиваем имена переменным.

Переменная в алгоритме	Имя на Фортране
x	X
$в$	B
a_i	A(I)
s	S
p	P
z	Z
y	Y
n	N

4. Выбираем спецификации вводимых данных.

Переменная	Пробивка на перфокарте	Спецификация
B	-1362	F5.2
X	3+2	E3.1
N	␣5	I2
A(I)	-12.6ØØØ	F8.4
	␣␣Ø.2461	
	␣-1.83ØØ	
	␣␣Ø.3ØØØ	
	175.1ØØØ	

5. Выбираем форму печати исходных данных и результата счета

Левый край бумаги

```
| 9поз ──────► ИВАНОВ␣А.␣С.
| 8поз ──────► ШИФР␣С␣75-2128
| 7поз ──────► ИСХОДНЫЕ␣ДАННЫЕ
|␣␣X=-13.62␣␣B=␣Ø.3E␣Ø2␣␣N=␣5
| 13поз ─────► A(I)
| 11поз ─────► -12.6ØØØ
|              ␣␣Ø.2461
|              и т.д.
| 10поз ─────► РЕЗУЛЬТАТ
| 9поз ──────► Y=␣O.ЦЦЦЕ␣ЦЦ
```

6. Записываем возможный вариант текста программы

```
     56
C     КОНТРОЛЬНАЯ␣РАБОТА␣N1
C     ЗАДАЧА 1
C     ИВАНОВ␣А.␣С.␣␣15.1Ø.1986.␣БПИ
      DIMENSION␣A(1Ø)
      READ(1,1) B,X,N,(A(I),I=1,N)
    1 FORMAT(F5.2,E3.1,I2/(F8.4))
      PRINT 2,X,B,N,(A(I),I=1,N)
    2 FORMAT(9X,'ИВАНОВ␣А.␣С.'/8X,
     *'ШИФР␣С␣75-2128'/7X,'ИСХОДНЫЕ␣ДАННЫЕ'/2X,
     *'X=',F6.2,2X,'B=',E8.1,2X,'N=',I2/13X,
     *'A(I)'/(11X,F8.4))
      Z=SQRT(ABS(B-X))/N
      S=Ø.
      P=1.
      DO␣3␣I=1,N
```

```
 5 6
      IF(A(I).GT.Ø.)S=S+A(I)
      IF(A(I).LT.Ø)P=P*A(I)
    3 CONTINUE
      IF(Z-Ø3*B)8,9,9
    8 Y=S+B
      GO TO 1Ø
    9 Y=P+X
   1Ø PRINT 15,Y
   15 FORMAT(1ØX,'РЕЗУЛЬТАТ'/9X,'Y=',E1Ø.5)
      STOP
      END
```

7. Записываем исходные данные

```
-13653+2 5
-12.6ØØØ
  Ø.2461
 -1.83ØØ
  Ø.3ØØØ
175.1ØØØ
```

Такую последовательность можно рекомендовать при решении задачи I контрольной работы № 2.

2.5. Программирование подпрограмм

2.5.I. Общие сведения

Рассмотрим основные правила написания подпрограмм на Фортране, из которых наиболее часто применяются подпрограммы типа *FUNCTION* и *SUBROUTINE*.

Для того, чтобы определить вычисления, которые будет производить подпрограмма-функция *FUNCTION*, следует написать участок программы, содержащий необходимые операторы, перед этим участком программы расположить заголовок, а в конце участка записать оператор *END*.

Начальная строка (заголовок) подпрограммы *FUNCTION* имеет вид:

ТИП *FUNCTION* ИМЯ ($a_1, \ldots, a_n$).

Здесь: ИМЯ – имя подпрограммы *FUNCTION*, присвоенное по обычным правилам. Оно состоит только из букв и цифр, начинается с буквы и содержит не более шести символов;

– ТИП – оператор описания типа подпрограммы – *INTEGER* (целый) или *REAL* (вещественный). Если этот оператор отсутствует, то тип подпрограммы определяется по первой букве имени;

– $a_1, \ldots, a_n$ – имена формальных параметров, разделяемые внутри скобок запятыми.

Формальные параметры являются фиктивными переменными и служат только для указания типа, количества и порядка записи аргументов функции. В качестве формальных параметров могут выступать имена переменных, массивов, подпрограмм.

Имя подпрограммы *FUNCTION* должно быть использовано в подпрограмме в качестве имени переменной в операторе присваивания.

Если в качестве формального параметра выступает имя массива, то оно должно быть описано внутри подпрограммы в операторе *DIMENSION*.

Подпрограмма *FUNCTION* должна содержать оператор *END* и не менее одного оператора *RETURN*. Оператор *END* указывает физический конец программы. Оператор *RETURN* означает логическое завершение вычислений и передает вычисленное значение функции в основную программу – ту программу, которая обращается к подпрограмме. Перед выполнением оператора *RETURN* имени функции должно быть присвоено вычисленное значение.

Обращение к подпрограмме *FUNCTION* производится путем указания в арифметическом выражении основной программы имени подпрограммы и следующего за ним списка фактических параметров, который заключается в круглые скобки. Параметры внутри списка разделяются запятыми. Формальные и фактические параметры должны соответствовать друг другу по типу, количеству и порядку следования. Выполнение обращения к подпрограмме *FUNCTION* приводит к тому, что значение функции вычисляется с использованием конкретных фактических параметров, значения которых заменяют формальные параметры. В качестве фактических параметров могут использоваться константы, переменные, массивы, элементы массивов, арифметические выражения, имена подпрограмм.

Подпрограмма *FUNCTION* имеет существенное ограничение, состоящее в том, что она всегда дает один явный результат. Этого недостатка лишена подпрограмма *SUBROUTINE*, которая может передавать в основную программу несколько результатов.

Заголовок подпрограммы *SUBROUTINE* имеет вид:

SUBROUTINE ИМЯ $(a_1, \ldots, a_n)$.

Здесь: ИМЯ – имя подпрограммы, присвоенное по обычным правилам. Однако оно не определяется по типу, так как при выполнении подпрограммы может быть получено несколько результатов. В отличие от подпрограммы *FUNCTION* имени подпрограммы *SUBROUTINE* не присваивается числовое значение;

– $a_1, \ldots, a_n$ – имена формальных параметров.

Формальные параметры и их соотношение с фактическими определяются так же, как и для подпрограммы *FUNCTION*.

Для обращения к подпрограмме из основной программы используется оператор *CALL*, записываемый в форме:

CALL ИМЯ $(b_1, \ldots, b_n)$.

Здесь: ИМЯ – имя подпрограммы *SUBROUTINE*;

$b_1, \ldots, b_n$ – фактические параметры.

Если имя некоторой внешней подпрограммы используется в качестве фактического параметра при обращении к данной подпрограмме, то оно должно быть описано в операторе *EXTERNAL* основной программы. Общая форма этого оператора

$$EXTERNAL\ a_1, \ldots, a_n,$$

где a_i – имена внешних подпрограмм, используемых в качестве фактических параметров.

Оператор *EXTERNAL* располагается в начале основной программы.

2.5.2. Пример разработки подпрограммы

Постановка задачи

Составить подпрограмму *SUBROUTINE* для формирования вектора

(b_i) из отрицательных элементов вектора (a_i). Используя эту подпрограмму, сформировать из отрицательных элементов вектор (y_i); выбрав из 10-ти элементов вектора (x_i) элементы, имеющие отрицательное численное значение.

Решение

1. Назовем подпрограмму именем *OTRWEK*.

2. Для получения универсальной подпрограммы зададим переменную размерность векторов входного – (a_i) и выходного (b_i). Схема алгоритма может иметь следующий вид:

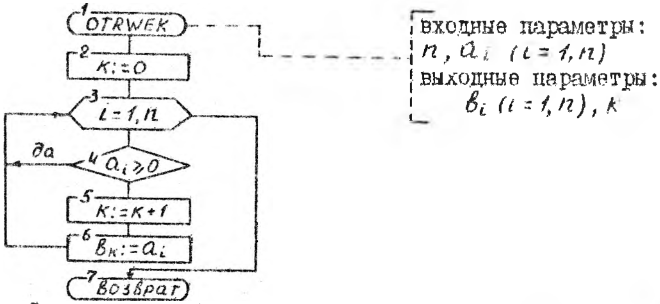
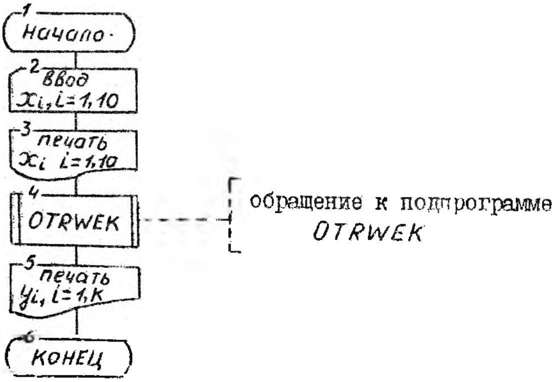


Схема головной программы, использующей подпрограмму *OTRWEK*, может иметь вид



3. Присваиваем имена переменным

Переменные в алгоритмах	a_i	b_i	x_i	y_i	i	k	n
Имена в программах	A(I)	B(I)	X(I)	Y(I)	I	K	N

4. С целью отладки подпрограммы присвоим элементам массива головной программы следующие значения:

X(I) = I,5	X(6) = 6,3
X(2) = -0,8	X(7) = -0,5
X(3) = -3,6	X(8) = -9,I
X(4) = 6,9	X(9) = 7,3
X(5) = -2,7	X(I0)= 9,5 ,

введя их с двух перфокарт по формату 5F4.I.

5. Результат выполнения программы распечатаем на АЦПУ в виде

Левый край бумаги

```
4поз  ИВАНОВ ␣ А. ␣ С.
3поз  ШИФР ␣ С. 75 - 2128
␣␣ ИСХОДНЫЙ ␣␣ МАССИВ
4поз      X(␣ 1) = ␣ 1.5
          X(␣2) = - Ø.8
          . . . . . . .
          X(1Ø) = ␣ 9.5
␣ СФОРМИРОВАННЫЙ ␣ МАССИВ
          Y(␣1)= -Ø.8
          Y(␣2)= -3.6
          Y(␣3)= -2.7      и т.д.
```

6. Записываем текст подпрограммы

```
     5 6
C        ПОДПРОГРАММА ␣ OTRWEK
C        КОНТРОЛЬНАЯ ␣ РАБОТА ␣ N2
C        ЗАДАЧА ␣ 2
C        ИВАНОВ ␣ А. ␣ С. ␣␣ 15.1Ø.1986. ␣ БПИ
         SUBROUTINE ␣ OTRWEK(A,N,B,K)
         DIMENSION ␣ A(N), B(N)
         K = Ø
         DO ␣ 7 ␣ I = 1,N
         IF(A(I).GE.Ø.)GO ␣ TO ␣ 7
         K = K+1
         B(K)=A(I)
      7  CONTINUE
         RETURN
         END
```

Обращение к подпрограмме *OTRWEK* :

CALL ␣ OTRWEK (A, N, B, K),

где A – входной вектор размерности N ;

B – выходной вектор размерности N ;

N – размерность векторов A и B ;

K – выходное значение максимального порядкового номера отрицательного элемента в векторе B

7. Записываем текст головной программы

```
     5 6
C       КОНТРОЛЬНАЯ␣РАБОТА␣N2
C       ЗАДАНИЕ␣2
C       ИВАНОВ ␣А.␣С.␣15.1Ø.1986.␣БПИ
C       ОТЛАДОЧНАЯ␣ПРОГРАММА␣К
C       ПОДПРОГРАММЕ␣OTRWEK
        DIMENSION␣X(1Ø),Y(1Ø)
        READ(*,1) (X(I),I=1,1Ø)
      1 FORMAT (5F4.1)
        PRINT␣2,(I,X(I),I=1,1Ø)
      2 FORMAT (4X,'ИВАНОВ␣А.␣С.'/3X,
     *'ШИФР␣С␣75-2178'/2X,
     *'ИСХОДНЫЙ␣␣МАССИВ'/,4X,'X(',I2,')=',F4.1))
        CALL␣OTRWEK (X,1Ø,Y,K)
        PRINT␣8,(I,Y(I),I=1,K)
      8 FORMAT(1X,'СФОРМИРОВАННЫЙ␣МАССИВ'/(4X,
     *'Y(',I2,')=',F4.1))
        STOP
        END
```

Исходные данные к этой программе должны быть записаны на бланке следующим образом:

␣1.5-Ø.8-3.6␣6.9-2.7
␣6.3-0.5-9.1␣7.3␣9.5

Такую последовательность можно рекомендовать при решении задачи 2 контрольной работы 2.

2.6. Использование пакета научных подпрограмм для решения прикладных задач

2.6.1. Общие сведения

В практике научных и инженерных расчетов часто возникает необходимость в решении таких задач, как интегрирование функций, решение линейных и нелинейных алгебраических, дифференциальных уравнений и их систем и т.п.. Численные методы решения таких задач разработаны математиками, запрограммированы и включены в состав программного обеспечения ЭВМ. Программы, реализующие эти методы, оформлены как подпрограммы. Для решения конкретной задачи на ЭВМ достаточно воспользоваться одной из этих подпрограмм, обращение к которым осуществляется стандартным образом. Использование таких подпрограмм, названных научными подпрограммами, обеспечивает возможность решения на ЭВМ задач из различных областей науки и техники, таких как физика, энергетика, экономика, химия, биология и др.при минимальных затратах времени на программирование.

Научные подпрограммы решения прикладных задач, написанные на Фортране, оформлены как подпрограммы FUNCTION и SUBROUTINE. При их использовании необходимо выдерживать приведенные выше правила работы с программами.

2.6.2. Решение систем линейных алгебраических уравнений

Рассмотрим систему n линейных алгебраических уравнений с n неизвестными

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{array}\right] -$$

или в векторно-матричной форме

$$Ax = b,$$

где

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}, \quad b = \begin{vmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{vmatrix}, \quad x = \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{vmatrix}$$

– соответственно матрица коэффициентов, вектор-столбец свободных членов и вектор-столбец неизвестных.

Для решения таких систем уравнений на практике широко применяется метод Гаусса с выбором главного элемента. В пакете научных подпрограмм этот метод реализован подпрограммой *SIMQ*, обращение к которой осуществляется так:

```
CALL SIMQ(A,B,N,IER).
```

Входные параметры: *N* – количество уравнений системы линейных алгебраических уравнений; *A* – матрица коэффициентов системы размером ($N \times N$), расположенная по столбцам. В ходе вычислений изменяется; *B* – вектор свободных членов системы, содержащий *N* элементов. На выходе заменяется вектором решения *X*.

Выходные параметры: *B* – массив из *N* действительных чисел, содержащий решение системы; *IER* – признак результата, принимающий следующие значения. *IER* = 0 – для нормального решения, *IER* = 1 – для особенной системы уравнений (определитель системы равен нулю).

2.6.3. Решение нелинейных уравнений

В практике научных и инженерных расчетов часто возникает необходимость в решении уравнений вида

$$f(x) = 0,$$

где функция $f(x)$ определена и непрерывна на некотором конечном или бесконечном интервале $a \le x \le b$. Численные методы решения таких уравнений позволяют определить приближенные значения корней с заданной степенью точности. На практике находят применение методы деления пополам и последовательных приближений.

Для метода деления пополам существенно, чтобы функция $f(x)$ была непрерывна и ограничена в заданном интервале $[a, b]$, внутри которого ищется корень. Предполагается также, что значения функции на концах интервала $f(a)$ и $f(b)$ имеют разные знаки, т.е. выполняется условие $f(a)\cdot f(b)<0$.

В пакете научных подпрограмм метод деления пополам обеспечивает подпрограмма RTILI. Обращение к данной подпрограмме:

CALL RTILI(X,F,FCT,XL,XR,EPS,MI,IER).

Входные параметры: FCT – имя внешней подпрограммы-функции. Эта подпрограмма составляется пользователем и должна обеспечить вычисление значения функции $f(x)$ по заданному значению x. Параметр FCT описывается оператором EXTERNAL в головной программе; XL, XR – соответственно значения левой и правой границ интервала $[a,b]$; EPS – погрешность вычисления корня; MI – максимальное число заданных итераций.

Выходные параметры: X – вычисленный корень уравнения $f(x)=0$; F – значение функции $f(x)$ в корне x; IER – признак результата, принимающий одно из следующих значений. IER= 0 – нет ошибки. IER =I – нет сходимости за MI итераций. IER =2 – не выполняется условие применимости метода деления пополам $FCT(XL)\cdot FCT(XR)\le 0$.

В методе последовательных приближений уравнение $f(x)=0$ представляют в виде $x=\varphi(x)$, что всегда можно сделать и притом многими способами. Например, можно выделить x из уравнения $f(x)=0$, а остальное перенести в правую часть. Можно также выполнить следующее преобразование:

$$x=x+Cf(x),$$

где C – произвольная постоянная. В этом случае $\varphi(x)=x+Cf(x)$.

Задаются начальным приближением $x[0]$, а последующие приближения определяют итерационной процедурой вида:

$$x[k+1]=\varphi(x[k]),\quad k=0,1,2,\ldots$$

Метод последовательных приближений реализует подпрограмма RTWI.

Обращение к подпрограмме:

CALL RTWI(X,F,FCT,XST,EPS,MI,IER).

Входные параметры: FCT – имя внешней подпрограммы-функции. Она составляется пользователем и должна обеспечить вычисление значения функции $\varphi(x)$ по заданному значению x. Параметр FCT в головной программе описывается оператором EXTERNAL; XST – начальное приближение корня; EPS – верхняя граница погрешности вычисления корня;

MI – максимальное число итераций.

Выходные параметры: X – вычисленный корень уравнения $x = \varphi(x)$; F – значение функции $\varphi(x)$ в корне x ; IER –признак результата, принимающий одно из следующих значений. IER=0 – нет ошибки. IER=I – нет сходимости за МI итераций. IER=2 – на некотором шаге знаменатель итерационной формулы равен нулю.

2.6.4. Интерполирование функций

На практике в результате экспериментальных исследований часто получают таблицу значений некоторой функции $f(x)$ при фиксированных значениях аргумента x , т.е. $f(x_i)$ $(i = 0, 1, \dots, n)$. Аналитическая взаимосвязь между x_i и $f(x_i)$ неизвестна, что не позволяет вычислить значения функции $f(x)$ в промежуточных точках, отличающихся от экспериментальных точек x_i $(i = 0, 1, \dots, n)$. Для описания этих значений строят аппроксимирующую (приближающую) функцию $\varphi(x)$, расчеты по которой, либо совпадают, либо в некотором смысле приближаются к экспериментально наблюдаемым значениям. Построение функции $\varphi(x)$ называется интерполированием. К интерполированию прибегают и в случае, когда аналитический вид функции $f(x)$ известен, но для получения ее значений необходимо провести большой объем вычислений. Замена функции $f(x)$ приближенной формулой $\varphi(x)$ позволяет упростить вычисления.

Наиболее часто в качестве интерполирующих функций $\varphi(x)$ применяют алгебраические многочлены. Интерполяция в таком случае называется алгебраической. На практике для интерполирования функций находят применение методы Лагранжа и Ньютона.

В пакете научных подпрограмм метод Лагранжа осуществляет подпрограмма ALI . Обращение к подпрограмме

CALL ALI (X, AX, AY, Y, N, EPS, IER).

Входные параметры: X – значение аргумента, для которого вычисляется приближенное значение функции; AX – вектор значений аргумента $x_1, \dots, x_n$ разномерности N ; AY – вектор экспериментальных значений функции $f(x_1), \dots, f(x_n)$ размерности N ; N – размерность векторов AX и AY ; EPS – верхняя граница абсолютной ошибки.

Выходные параметры: Y – интерполированное значение функции; IER – признак результата. Этот параметр принимает одно из четырех значений. IER= 0 – требуемая точность достигается. В данном случае разность между двумя последовательными интерполированными значениями функции по модулю меньше заданного значения EPS (проверка начинается с третьего шага). IER = I – требуемая точность не достигается вследствие округления. $IER = 2$ – $N<3$. IER = 3 – обнаружено два одинаковых значения аргумента в векторе АХ.

Линейная интерполяция по методу Ньютона реализована в подпрограмме $LINTX$. Обращение к подпрограмме:

$$CALL\ LINTX(X, Y, AX, AY, N, IER).$$

Входные параметры: Х – значение аргумента, для которого вычисляется приближенное значение функции; АХ – массив значений аргумента $x_1, \ldots, x_n$; АУ – массив экспериментальных значений функции $f(x_1), \ldots, f(x_n)$; N – размерность массивов АХ, АУ.

Выходные параметры: У – интерполированное значение функции; IER – признак результата, принимающий одно из следующих значений. IER = 0 – значение Х находится в пределах таблицы [АХ(I), АХ(N)], решение получено. IER=I – значение аргумента Х выходит за пределы таблицы [AX (I), $AX(N)$], вычисление Y не производится.

2.6.5. Численное интегрирование

На практике для вычисления значений определенных интегралов находят применение методы трапеции и Симпсона.

Вычисление вектора значений интегралов $Z_i = \int_a^{x_i} f(x)\,dx$ для таблицы значений функции $f(x_i)$, заданной в равностоящих точках $x_i = a + (i-1)h\ (i = 1, 2, \ldots, n)$, по правилу трапеций осуществляет подпрограмма $QTFE$. Начиная со значения $Z_1 = 0$, вычисляются последовательные значения интегралов.

$$Z_i = Z_{i-1} + \frac{h}{2}\left(f(x_i) + f(x_{i-1})\right), \quad i = 2, \ldots, n$$

Обращение к подпрограмме

$$CALL\ QTFE(H, Y, Z, N).$$

Входные параметры: H – приращение значений аргумента, Y – вектор значений функции, N – размерность векторов Y, Z.

Выходной параметр: Z – вектор значений интегралов.

Вычисление вектора значений интегралов для табличной функции с равноотстоящими значениями аргумента по формуле Симпсона реализовано в подпрограмме QSF. Обращение к подпрограмме

$$CALL\ QSF(H, Y, Z, N).$$

Входные параметры: H – приращение значений аргумента, Y – вектор значений функции, N – размерность векторов Y, Z. Если $N<3$, то вычисления не производятся.

Выходной параметр: Z – вектор значений интегралов.

2.6.6. Решение обыкновенных дифференциальных уравнений

На практике для решения дифференциальных уравнений первого порядка

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

с начальными условиями $y_0 = y(x_0)$ широкое применение находят методы Рунге-Кутта. Один из этих методов реализован в пакете научных подпрограмм в подпрограмме $RK2$. ЭВМ выдает табличную зависимость значений y от значений x. Обращение к подпрограмме имеет вид

```
CALL RK2(FUN, H, X1, Y1, K, N, VEC).
```

Входные параметры: FUN – имя подпрограммы $FUNCTION$ для вычисления значений $f(x,y)$, составляемой пользователем. Параметр FUN описывается оператором $EXTERNAL$; H – шаг интегрирования; $X1$, $Y1$ – соответственно начальные значения x_0, y_0; К – количество шагов между двумя сохраняемыми в памяти величинами; N – число накапливаемых значений.

Выходные параметры: VEC – результирующий вектор длины N, в котором накапливаются вычисленные значения Y.

В подпрограмме вычисляются значения Y для значений X, расстояние между которыми равно шагу H, но в памяти сохраняется каждое К-е значение Y (для значений X, расстояние между которыми составляет KH). Количество накапливаемых в массиве VEC значений Y равно

$$N = \frac{X_K - X_H}{KH}.$$

Здесь X_H, X_K – соответственно начальное и конечное значение интервала, на котором решается уравнение.

2.6.7. Пример решения задачи с использованием научных подпрограмм

При выполнении контрольной работы № 3 необходимо учесть следующее обстоятельство. В каждом из вариантов имеются две задачи. В одной из них требуется написать основную программу с обращением к подпрограмме, реализующей соответствующий метод решения конкретной задачи. В другой необходимо написать основную программу с обращением к подпрограмме, реализующей конкретный метод решения задачи, и, кроме того, подпрограмму типа *FUNCTION* для вычисления значений функций, предусмотренных условиями задания. Проиллюстрируем методику составления программ для решения прикладных задач с использованием научных подпрограмм на примере решения задачи второго типа как более сложной. При решении задачи будем учитывать требования контрольной работы.

Пример. Для вычисления длины кривой, являющейся графиком функции $y=f(x)$, данную кривую аппроксимируют ломаной линией и вычисляют длину этой линии. Такой метод реализован в подпрограмме *LONG*, обращение к которой осуществляется следующим образом:

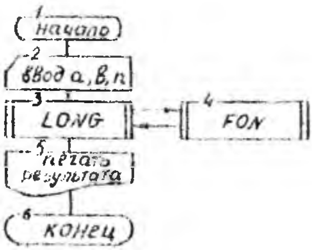
$$CALL\ LONG(F,A,B,N,S).$$

Входные параметры: F – имя внешней подпрограммы *FUNCTION*, составляемой пользователем для вычисления значений функции $f(x)$. Параметр F описывается оператором *EXTERNAL*; A – нижний предел изменения аргумента; B – верхний предел изменения аргумента, N – число интервалов, на которые делится отрезок $[A,B]$ при аппроксимации функции.

Выходной параметр: S – вычисленное значение длины кривой.

Используем эту подпрограмму для вычисления длины кривой, соответствующей функции $y=\sqrt{x(2\emptyset-x)}$. Значение x изменяется от $a=\emptyset$ до $b=2\emptyset$, число интервалов $n=5\emptyset$. Для вычисления значений функции $f(x)$ разработаем внешнюю подпрограмму *FUNCTION* с именем *FON*.

1. Вычерчиваем схему алгоритма решения задачи.



2. Присваиваем имена переменным.

Переменная	Имя на Фортране
d	D
a	A
b	B
n	N

3. Выбираем спецификации вводимых данных.

Величины b, a в общем случае являются вещественными, поэтому выбираем для них спецификацию F3.1. Для числа n используем спецификацию I.2. Все числа введем с одной перфокарты.

4. Выбираем форму печати результатов.

```
                 30 позиций
Левый  |______________________________ИВАНОВ␣А.С.
край   |                              ШИФР␣С␣75-2128
бумаги |                              ДЛИНА␣КРИВОЙ␣S= ____________
                                                      I2 позиций
```

Для печати результата используем спецификацию E12.5.

5. Записываем текст программы.

```
     56
C     КОНТРОЛЬНАЯ␣РАБОТА␣N3
      EXTERNAL FON
      READ(1,1)A,B,N
    1 FORMAT(2F3.1,I2)
      CALL LONG(FON,A,B,N,S)
      WRITE(3,2)S
    2 FORMAT(3ØX,1ØHИВАНОВ␣А.С/3ØX,
     *14HШИФР␣С␣75-2128/3ØX,5HДЛИНА,1X,
     *9HКРИВОЙ␣S=,F12.5)
      STOP
      END
      FUNCTION FON(X)
      FON=SQRT(X*(2Ø.-X))
      RETURN
      END
Ø.Ø2Ø.5Ø
```

С О Д Е Р Ж А Н И Е

Феликс Михайлович БАБУШКИН
Осип Викентьевич БУГАЙ
Владимир Владимирович ВЫГОВСКИЙ и др.

ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению контрольных работ по курсу
"Вычислительная математика и программирование на ЭВМ"
для студентов специальностей 0502, 0503
заочной формы обучения

Редактор Н.Я.Пронина

Подписано в печать 26.01.87.
Формат 60х84 1/16. Бумага т.№2. Офс.печать.
Усл.печ.л.2.56. Уч.-изд.л.2.00. Тир.800. Зак.58. Бесплатно.

Отпечатано на ротапринте БПИ. 220027, Минск, Ленинский пр., 65.