Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
(ТУСУР)

Н.В. Зариковская

МОДЕЛИРОВАНИЕ СИСТЕМ

Учебно-методическое пособие для аудиторных практических занятий, лабораторных работ и самостоятельных работ студентов

2018

Зариковская Н.В.
МОДЕЛИРОВАНИЕ СИСТЕМ
Учебно-методическое пособие для аудиторных практических занятий, лабораторных работ и самостоятельных работ студентов. – Томск: Томский государственный университет систем управления и радиоэлектроники (ТУСУР), 2018. – 103 с.

СОДЕРЖАНИЕ

1. Практические занятия

Практические занятия предназначены для закрепления лекционного материала и предусматривают решение задач и примеров. По лекционному материалу сформированы девять практических работ распределенных на пять разделов:

1. Основные этапы решения инженерных задач. Вычисление погрешностей.
2. Решение уравнений с одной переменной.
3. Задачи линейной алгебры. Вычисление собственных чисел и собственных векторов.
4. Интерполирование и численное дифференцирование функций. Приближение сплайнами.
5. Численное интегрирование. Решение дифференциальных уравнений.

При решении практических заданий необходимо использование изученных на лекциях численных методов.

1.1 Общие указания по выполнению работ

1.1.1 Практическая работа № 1

Работа № 1 выполняется после изучения глав «Приближенные числа», «Погрешности арифметических действий» и «Приближенное решение алгебраических уравнений».

Пример решения типового варианта

Задача 1

Представить число $7642.541....$ в виде бесконечной десятичной дроби.

Решение. Чтобы представить число в виде бесконечной десятичной дроби, необходимо разложить его по степеням числа 10.

$$7642.541... = 7 \cdot 10^3 + 6 \cdot 10^2 + 4 \cdot 10^1 + 2 \cdot 10^0 +$$

$$+5 \cdot 10^{-1} + 4 \cdot 10^{-2} + 1 \cdot 10^{-3} + ...$$

Задача 2

Определить значащие цифры чисел $a1 = 0.00439$; $a2 = 2.8065$; $a3 = 12.0057035$; $a4=5.72 \cdot 10^4$; $a5=5.730 \cdot 10^3$.

Решение. Значащими цифрами числа являются все цифры в его десятичном изображении, отличные от нуля, и нуль, если он содержится между значащими цифрами или является представителем сохраненного десятичного разряда.

Поэтому в числе $a1 = 0.00439$ первые три нуля не являются значащими (служат для установления десятичных разрядов следующих цифр), в числах $a2 = 2.8065$ и $a3 = 12.0057035$ все цифры - значащие (все нули содержатся между значащими цифрами), в числе $a4=5.72 \cdot 10^4$ значащими являются первые три цифры (нули не являются представителями сохраненных десятичных разрядов), в числе $a5=5.730 \cdot 10^3$ значащими являются четыре цифры.

Задача 3

Округлить число 4.12657 до тысячных, до сотых, до десятых. Найти абсолютную и относительную погрешность каждого результата.

Решение. Пользуясь правилами округления, получим:

при округлении до тысячных – 4.127;

при округлении до сотых – 4.13;

при округлении до десятых – 4.1.

Поскольку абсолютная погрешность округления не превосходит $\frac{1}{2}$ единицы десятичного разряда, определяемого последней оставленной значащей цифрой, то $\Delta_1 < 0.5 \cdot 10^{-3}$, $\Delta_2 < 0.5 \cdot 10^{-2}$, $\Delta_3 < 0.5 \cdot 10^{-1}$.

Относительная погрешность каждого из результатов:

$$\delta_1 < \frac{0.5 \cdot 10^{-3}}{4} = 0.00012, \qquad \delta_2 < \frac{0.5 \cdot 10^{-2}}{4} = 0.0012,$$

$$\delta_3 < \frac{0.5 \cdot 10^{-1}}{4} = 0.012.$$

Задача 4

Со сколькими знаками надо взять число $\ln 27$, чтобы относительная погрешность была не больше 0.1%.

Решение. Воспользуемся формулой для нахождения числа знаков по известной относительной погрешности:
$n=[1-\lg(\alpha_m \cdot \delta)]$. (Здесь квадратные скобки обозначают округление результата до ближайшего целого, величина α_m - первая значащая цифра результата).

Поскольку $\ln 27 \approx 3.29$, то $\alpha_m = 3$ и $n=[1-\lg(3 \cdot 0{,}001)]=4$. Таким образом, для выполнения заданной точности, число $\ln 27$ нужно взять с 4 знаками: $\ln 27=3.29583=3.286$.

Задача 5

При измерении длины участка пути в 7 км допущена ошибка в 2 м, а при измерении длины болта в 7 см допущена ошибка в 2 мм. Какое из этих измерений более точное?

Решение. Необходимо найти и сравнить относительные погрешности каждого измерения.

δ_1 =2м/7000м=$2.85 \cdot 10^{-4}$;

δ_2 =2мм/70мм=$2.85 \cdot 10^{-2}$.

Таким образом, боле точным является измерение участка пути, поскольку относительная погрешность этого измерения меньше.

1.1.2 Практическая работа № 2

Работа № 2 выполняется после изучения глав «Приближенное решение нелинейных уравнений».

Пример решения типового варианта

Задача 6

1) Найти нуль функции методом Ньютона.

$$f(x) = x^3 - x + 1 \quad x \in [-2{,}0]$$

Выполнить две - три итерации.

Решение. Выберем x_0 из неравенства $f(x_0) \cdot f''(x_0) > 0$

$f' = 3x^2 - 1;$

$f'' = 6x.$

$x_0 = -2,\ f(-2) = -5,\ f''(-2) = -12;\ f(-2) \cdot f''(-2) = 60 > 0.$

$x_n = x_{n-1} - f(x_{n-1}) / f'(x_{n-1})$

N	x	$f(x)$	$f'(x)$
0	-2	-5	11
1	-1.55	-1.17	6.2
2	-1.362	-0.164	4.56

Ответ: $\xi = -1.362$

2) Найти методом хорд положительный корень с точностью до 0.002

$$f(x) = x^3 - 0.2x^2 - 0.2x - 1.2$$

Решение.

Определяем интервал, на котором находится корень.

Т.к. $f(1) = -0.6 < 0$, $f(2) = 5.6 > 0$ то $\xi \in [1, 2]$.

Вычисляем

$$x_1 = a - \frac{f(a)}{f(b) - f(a)}(b - a) = 1 - \frac{-0.6 \cdot 2}{5.6 - (-0.6)} \approx 1.194.$$

$f(1.194) = -0.022$. Так как $f(1.194) \cdot f(2) < 0$, то корень находится в интервале $[1.194, 2]$. Продолжим вычисления

$$x_2 = 1.194 - \frac{-0.022}{5.6 - (-0.022)}(2 - 1.194) = 1.197$$

Проверяем условие $|1.197 - 1.194| = 0.003 > \varepsilon$

Продолжаем процесс. $f(1.197) = -0.011$;

$$x_2 = 1.197 - \frac{-0.011}{5.6 - (-0.011)}(2 - 1.197) = 1.199;$$

$|1.199 - 1.197| = 0.002 = \varepsilon$. Процесс можно закончить.

Ответ: $\xi = 1.199$

3) Найти комбинированным методом корень уравнения $f(x) = x^5 - x - 0.2 = 0$ на интервале $[1, 1.1]$ с точностью $\varepsilon = 0.0005$.

Решение. Проверим наличие корня:

$f(1) = -0.2$; $f(1.1) = 0.3105$; $f(1) \cdot f(1.1) < 0$, следовательно корень существует. Вычислим производные

$f'(x) = 5x^4 - 1;\ f''(x) = 20x^3$. Итак на интервале $[1{,}1.1]$ $f'(x)$ и $f''(x)$ сохраняют знаки, причем $f'(x) > 0$ и $f''(x) > 0$. Далее, так как $f(1.1) \cdot f''(1.1) > 0$, то вычисления проводим по формулам (2.8): $x_0 = a = 1;\ b_0 = b = 1.1;\ f(b_0) = 6.3205$;

$$x_1 = x_0 - \frac{f(x_0)(b_0 - x_0)}{f(b_0) - f(x_0)} = 1 - \frac{-0.2(1.1-1)}{6.3205} = 1.03917;$$

$$b_1 = b_0 - \frac{f(b_0)}{f'(b_0)} = 1.1 - \frac{0.3105}{6.3205} = 1.05087.$$

Проверяем условие останова:

$|b_1 - x_1| = |1.05087 - 1.03917| = 0.0117 > \varepsilon$; точность не достигнута, продолжаем вычисления. $f(x_1) = f(1.03917) = -0.02736$; $f(b_1) = f(1.05087) = 0.0307$; $f'(b_1) = f'(1.05087) = 5.0977$;

$$x_2 = x_1 - \frac{f(x_1)(b_1 - x_1)}{f(b_1) - f(x_1)} = 1 - \frac{-0.02736}{0.0307 - (-0.02736)}(1.05087 - 1.03917) = 1.04468;$$

$$b_2 = b_1 - \frac{f(b_1)}{f'(b_1)} = 1.05087 - \frac{0.0307}{5.0977} = 1.04485.$$

$|b_2 - x_2| = |1.04485 - 1.04468| = 0.0002 < \varepsilon;$

Ответ: $\xi = \dfrac{1}{2}(b_2 + b_2) = 1.045$.

1.1.3 Практическая работа № 3

Работа № 3 выполняется после изучения глав «Численные методы линейной алгебры» и «Приближенное решение систем нелинейных уравнений».

Пример решения типового варианта

Задача 1. Решить систему методом Гаусса

$$\begin{cases} 7x_1 + 2x_2 + 3x_3 = 15 \\ 5x_1 - 3x_2 + 2x_3 = 15 \\ 10x_1 - 11x_2 + 5x_3 = 36 \end{cases}$$

Решение. Результаты прямого хода выпишем в таблицу

	x_1	x_2	x_3	b
A	7	2	3	15
	5	-3	2	15
	10	-11	5	36
A_1	1	2/7	3/7	15/7
	0	-31/7	-1/7	30/7
	0	-97/7	5/7	102/7
A_2	1	2/7	3/7	15/7
	0	1	1/31	-30/31
	0	0	36/31	36/31
A_3	1	2/7	3/7	15/7
	0	1	1/31	-30/31
	0	0	1	1

Обратный ход

$$x_3 = 1; \quad x_2 = -\frac{30}{31} - \frac{1}{31} \cdot 1 = -1; \quad x_1 = \frac{15}{7} - \frac{2}{7} \cdot (-1) - \frac{3}{7} \cdot 1 = 2 \ .$$

Задача 2

Найти собственные числа и собственные вектора матрицы.

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 2 & 3 \\ 3 & 2 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix}$$

Решение. Вычисляем матрицу $\tilde{A}_1$ по следующим формулам

$$\tilde{a}^1{}_{13} = \frac{a_{13}}{a_{43}} = \frac{3}{2}; \quad \tilde{a}^1{}_{23} = \frac{a_{23}}{a_{43}} = 1; \quad \tilde{a}^1{}_{33} = \frac{a_{33}}{a_{43}} = \frac{1}{2}; \quad \tilde{a}^1{}_{23} = \frac{a_{43}}{a_{43}} = 1;$$

$$\tilde{a}^1{}_{11} = a_{11} - a_{41}\frac{a_{13}}{a_{43}} = 1 - 4\cdot\frac{3}{2} = -5; \quad \tilde{a}^1{}_{21} = a_{21} - a_{41}\frac{a_{23}}{a_{43}} = -2;$$

$$\tilde{a}^1{}_{31} = a_{31} - a_{41}\frac{a_{33}}{a_{43}} = 3 - 4\cdot\frac{1}{2} = 1; \quad \tilde{a}^1{}_{41} = a_{41} - a_{41}\frac{a_{43}}{a_{43}} = 0;$$

$$\tilde{a}^1{}_{12} = a_{12} - a_{42}\frac{a_{13}}{a_{43}} = 2 - 3\cdot\frac{3}{2} = -\frac{5}{2}; \quad \tilde{a}^1{}_{22} = a_{22} - a_{42}\frac{a_{23}}{a_{43}} = -2;$$

$$\tilde{a}^1{}_{32} = a_{32} - a_{42}\frac{a_{33}}{a_{43}} = 2 - 3\cdot\frac{1}{2} = \frac{1}{2}; \quad \tilde{a}^1{}_{42} = a_{42} - a_{42}\frac{a_{43}}{a_{43}} = 0;$$

$$\tilde{a}^1{}_{14} = a_{14} - a_{44}\frac{a_{13}}{a_{43}} = 4 - 1\cdot\frac{3}{2} = \frac{5}{2};$$

$$\tilde{a}^1{}_{24} = a_{24} - a_{44}\frac{a_{23}}{a_{43}} = 3 - 1\cdot 1 = 2;$$

$$\tilde{a}^1{}_{34} = a_{34} - a_{44}\frac{a_{33}}{a_{43}} = 2 - 1\cdot\frac{1}{2} = \frac{3}{2}; \quad \tilde{a}^1{}_{44} = a_{44} - a_{44}\frac{a_{43}}{a_{43}} = 0.$$

Таким образом, матрица $\tilde{A}_1$ равна

$$\tilde{A}_1 = \begin{pmatrix} -5 & -5/2 & 3/2 & 5/2 \\ -2 & -2 & 1 & 2 \\ 1 & 1/2 & 1/2 & 3/2 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Далее вычисляем матрицу A_1. Она равна матрице $\tilde{A}_1$, в которой необходимо изменить только третью строку.

$$a^1_{31} = \sum_{k=1}^{4} a_{nk}\tilde{a}^1_{k1} = 4\cdot(-5) + 3\cdot(-2) + 2\cdot 1 + 0 = -24;$$

$$a^1_{32} = \sum_{k=1}^{4} a_{nk}\tilde{a}^1_{k2} = 4\cdot(-5/2) + 3\cdot(-2) + 2\cdot(1/2) + 0 = -15;$$

$$a_{33}^{1} = \sum_{k=1}^{4} a_{nk}\tilde{a}_{k3}^{1} = 4\cdot(3/2) + 3\cdot 1 + 2\cdot(1/2) + 1\cdot 1 = 11;$$

$$a_{34}^{1} = \sum_{k=1}^{4} a_{nk}\tilde{a}_{k4}^{1} = 4\cdot(5/2) + 3\cdot 2 + 2\cdot(3/2) + 0 = 19.$$

$$A_1 = \begin{pmatrix} -5 & -5/2 & 3/2 & 5/2 \\ -2 & -2 & 1 & 2 \\ -24 & -15 & 11 & 19 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Вычисляем матрицу $\tilde{A}_2$.

$$\tilde{a}_{12}^{2} = \frac{a_{12}^{1}}{a_{32}^{1}} = \frac{(-5/2)}{(-15)} = \frac{1}{6}; \quad \tilde{a}_{22}^{2} = \frac{a_{22}^{1}}{a_{32}^{1}} = \frac{(-2)}{(-15)} = \frac{2}{15}; \quad \tilde{a}_{32}^{2} = \frac{a_{32}^{1}}{a_{32}^{1}} = 1;$$

$$\tilde{a}_{11}^{2} = a_{11}^{1} - a_{31}^{1}\frac{a_{12}^{1}}{a_{32}^{1}} = -5 - (-24)\cdot\frac{1}{6} = -1;$$

$$\tilde{a}_{21}^{2} = a_{21}^{1} - a_{31}^{1}\frac{a_{22}^{1}}{a_{32}^{1}} = -2 - (-24)\cdot\frac{2}{15} = \frac{18}{15};$$

$$\tilde{a}_{31}^{2} = a_{31}^{1} - a_{31}^{1}\frac{a_{32}^{1}}{a_{32}^{1}} = 0;$$

$$\tilde{a}_{13}^{2} = a_{13}^{1} - a_{33}^{1}\frac{a_{12}^{1}}{a_{32}^{1}} = (3/2) - (11)\cdot(1/6) = -\frac{1}{3};$$

$$\tilde{a}_{23}^{2} = a_{23}^{1} - a_{33}^{1}\frac{a_{22}^{1}}{a_{32}^{1}} = 1 - (11)\cdot(2/15) = -\frac{7}{15};$$

$$\tilde{a}_{33}^{2} = a_{33}^{1} - a_{33}^{1}\frac{a_{32}^{1}}{a_{32}^{1}} = 0;$$

$$\tilde a^2_{14} = a^1_{14} - a^1_{34}\frac{a^1_{12}}{a^1_{32}} = (5/2) - (19)\cdot(1/6) = -\frac{2}{3};$$

$$\tilde a^2_{24} = a^1_{24} - a^1_{34}\frac{a^1_{22}}{a^1_{32}} = 2 - (19)\cdot(2/15) = -\frac{8}{15};$$

$$\tilde a^2_{34} = a^1_{34} - a^1_{34}\frac{a^1_{32}}{a^1_{32}} = 0.$$

$$\tilde A_2 = \begin{pmatrix} -1 & 1/6 & -1/3 & -2/3 \\ 18/15 & 2/15 & -7/15 & -8/15 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Вычисляем матрицу A_2 . Она равна матрице $\tilde A_2$, в которой необходимо изменить только вторую строку.

$$a^2_{21} = \sum_{k=1}^{4} a^1_{3k}\tilde a^2_{k1} = (-24)\cdot(-1) + (-15)\cdot(18/15) + 11\cdot 0 + 19\cdot 0 = 6;$$

$$a^2_{22} = \sum_{k=1}^{4} a^1_{3k}\tilde a^2_{k2} = (-24)\cdot(1/6) + (-15)\cdot(2/15) + 11\cdot 1 + 19\cdot 0 = 5;$$

$$a^2_{23} = \sum_{k=1}^{4} a^1_{3k}\tilde a^2_{k3} = (-24)\cdot(-1/3) + (-15)\cdot(-7/15) + 11\cdot 0 + 19\cdot 1 = 34;$$

$$a^2_{24} = \sum_{k=1}^{4} a^1_{3k}\tilde a^2_{k4} = (-24)\cdot(-2/3) + (-15)\cdot(-8/15) + 11\cdot 0 + 19\cdot 0 = 24.$$

$$A_2 = \begin{pmatrix} -1 & 1/6 & -1/3 & -2/3 \\ 6 & 5 & 34 & 24 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Аналогично вычисляем матрицу $\tilde A_3$.

$$\tilde{a}_{11}^3 = \frac{a_{11}^2}{a_{21}^2} = \frac{-1}{6}; \quad \tilde{a}_{21}^3 = \frac{a_{21}^2}{a_{21}^2} = 1;$$

$$\tilde{a}_{12}^3 = a_{12}^2 - a_{22}^2 \frac{a_{11}^2}{a_{21}^2} = (1/6) - 5 \cdot (-1/6) = 1; \quad \tilde{a}_{22}^3 = 0;$$

$$\tilde{a}_{13}^3 = a_{13}^2 - a_{23}^2 \frac{a_{11}^2}{a_{21}^2} = (-1/3) - 34 \cdot (-1/6) = 16/3; \quad \tilde{a}_{23}^3 = 0;$$

$$\tilde{a}_{14}^3 = a_{14}^2 - a_{24}^2 \frac{a_{11}^2}{a_{21}^2} = (-2/3) - 24 \cdot (-1/6) = 10/3; \quad \tilde{a}_{24}^3 = 0.$$

$$\tilde{A}_3 = \begin{pmatrix} -1/6 & 1 & 16/3 & 10/3 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Вычисляем матрицу A_3 по формулам.

$$\begin{cases} a^{n-1}{}_{1,j} = \sum_{k=1}^{n} a^{n-2}{}_{1,k} \tilde{a}^{n-1}{}_{kj}; \\ j = 1, \ \ldots, \ n \end{cases}$$

$$a_{11}^3 = \sum_{k=1}^{4} a_{1k}^2 \tilde{a}_{k1}^3 = 6 \cdot (-1/6) + 5 \cdot 1 = 4;$$

$$a_{12}^3 = \sum_{k=1}^{4} a_{1k}^2 \tilde{a}_{k2}^3 = 6 \cdot 1 + 5 \cdot 0 + 34 \cdot 1 = 40;$$

$$a_{13}^3 = \sum_{k=1}^{4} a_{1k}^2 \tilde{a}_{k3}^3 = 6 \cdot (16/3) + 5 \cdot 0 + 34 \cdot 0 + 24 \cdot 1 = 56;$$

$$a_{14}^3 = \sum_{k=1}^{4} a_{1k}^2 \tilde{a}_{k4}^3 = 6 \cdot (10/3) = 20.$$

В результате мы получили следующую матрицу Фробениуса

$$P = A_3 = \begin{pmatrix} 4 & 40 & 56 & 20 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Запишем характеристический полином

$$D(\lambda) = \begin{vmatrix} 4-\lambda & 40 & 56 & 20 \\ 1 & -\lambda & 0 & 0 \\ 0 & 1 & -\lambda & 0 \\ 0 & 0 & 1 & -\lambda \end{vmatrix}$$

или $D(\lambda) = \lambda^4 - 4\lambda^3 - 40\lambda^2 - 56\lambda - 20$. Корни равны:

$$\lambda_1 = 9.098975;$$

$$\lambda_2 = -0.585791;$$

$$\lambda_3 = -1.098975;$$

$$\lambda_4 = -3.414209.$$

Вычислим собственный вектор для $\lambda_2 = -0.585791.$.

- Вычисляем собственный вектор матрицы Фробениуса

$$y = (\lambda^3,\ \ \lambda^2,\ \ \lambda,\ 1) = (-0.20101,\ \ 0.34315,\ \ -0.58579,\ \ 1).$$

- Вычисляем вектор y^1

$$m_{11} = \frac{1}{a_{21}^2};\ m_{1k} = -\frac{a_{2k}^2}{a_{21}^2};\quad k \neq 1.$$

Из элементов матрицы A_2 определяем

$$m_{11} = 1/6;\quad m_{12} = -5/6;\quad m_{13} = -34/6;\quad m_{14} = -24/6;$$

$$y_1^1 = \sum_{k=1}^{n} m_{1k} y_k^0 = (1/6)\cdot[1\cdot(-0.20101) + (-5)\cdot 0.34315 + (-34)\cdot$$

$$\cdot(-0.58579) + (-24)\cdot 1] = 0.99998;$$

$$y^1 = (0.99998,\ \ 0.34315,\ \ -0.58579,\ \ 1).$$

- Вычисляем вектор y^2 .

Из элементов матрицы A_1 определяем

$$m_{21} = -\frac{a_{31}^1}{a_{32}^1} = -\frac{24}{15}; \quad m_{22} = \frac{1}{a_{32}^1} = -\frac{1}{15};$$

$$m_{23} = -\frac{a_{33}^1}{a_{32}^1} = \frac{11}{15}; \quad m_{24} = -\frac{a_{34}^1}{a_{32}^1} = \frac{19}{15};$$

$$y_2^2 = \sum_{k=1}^{n} m_{2k} y_k^1 = (1/15)\cdot[(-24)\cdot(-0.99998) +$$

$$(-1)\cdot 0.34315 + 11\cdot\cdot(-0.58579) + 19\cdot 1] = 2.41418;$$

$$y^2 = (-0.99998,\ \ 2.41418,\ \ -0.58579,\ \ 1).$$

- Вычисляем вектор y^3.

Из элементов матрицы $A_0 = A$ определяем

$$m_{31} = -\frac{a_{41}^0}{a_{43}^0} = -\frac{4}{2}; \quad m_{32} = -\frac{a_{42}^0}{a_{43}^0} = -\frac{3}{2};$$

$$m_{33} = \frac{1}{a_{43}^0} = \frac{1}{2}; \quad m_{32} = -\frac{a_{44}^0}{a_{43}^0} = -\frac{1}{2};$$

$$y_3^3 = \sum_{k=1}^{n} m_{3k} y_k^2 = (1/2)\cdot[(-4)\cdot(-0.99998) + (-3)\cdot 2.41418 + 1\cdot$$

$$\cdot(-0.58579) + (-1)\cdot 1] = -2.41420;$$

$$y^3 = (-0.99998,\quad 2.41418,\quad -2.41420,\quad 1).$$

Таким образом, собственному значению $\lambda_2 = -0.58579$ соответствует собственный вектор

$$x = y^3 = (-0.99998,\quad 2.41418,\quad -2.41420,\quad 1).$$

Проверка.

$$Ax = \lambda x .$$

Вычислим правую и левую части этого соотношения при $\lambda = \lambda_2 = -0.58579$.

$$\sum_{k=1}^{4} a_{1k} x_k = 0.58576; \qquad \lambda\cdot x_1 = 0.58578.$$

$$\sum_{k=1}^{4} a_{2k} x_k = -1.41418; \qquad \lambda\cdot x_2 = -1.41420.$$

$$\sum_{k=1}^{4} a_3 x_k = 1.41422; \qquad \lambda\cdot x_3 = -1.41421.$$

$$\sum_{k=1}^{4} a_4 x_k = -0.58578; \qquad \lambda\cdot x_4 = -0.58579.$$

Получили согласие в 4 цифрах после запятой.

Задача 3

Найти обратную матрицу, используя метод декомпозиции.

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 3 & 0 & 2 \\ 4 & -2 & 5 \end{pmatrix}$$

Решение: Нам надо решить три системы уравнений вида

$$Ax_i = e_i, \quad i = 1,\ 2,\ 3.$$

Будем решать эти системы методом декомпозиции. Представим матрицу A в виде произведения матриц: $A = B \cdot C$ и вычислим элементы матриц B и C

$b_{11} = a_{11} = 1; \quad b_{21} = 3; \quad b_{31} = 4;$

$c_{11} = 1; \quad c_{12} = 2; \quad c_{13} = -1;$

$b_{22} = a_{22} - b_{21}c_{12} = 0 - 3 \cdot 2 = -6;$

$b_{32} = a_{32} - b_{31}c_{12} = -2 - 4 \cdot 2 = -10;$

$$c_{22} = 1; \quad c_{23} = \frac{1}{b_{22}}(a_{23} - b_{21}c_{13}) =$$

$= 1/(-6) \cdot (2 - 3 \cdot (-1)) = -5/6;$

$b_{33} = a_{33} - b_{31}c_{13} - b_{32}c_{23} = 5 - 4 \cdot (-1) - (-10) \cdot (-5/6) =$

$= 2/3.$

Таким образом, получили

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 3 & -6 & 0 \\ 4 & -10 & 2/3 \end{pmatrix}; \quad C = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & -5/6 \\ 0 & 0 & 1 \end{pmatrix}.$$

Исходные системы раскладываются на две эквивалентные системы

$$By_i = e_i, \quad Cx_i = y_i, \quad i = 1,\ 2,\ 3.$$

Компоненты векторов y_i и x_i вычисляются по формулам

$y_{1i} = e_{1i} / b_{11}; \quad y_{2i} = (e_{2i} - b_{21}y_{1i}) / b_{22};$

$y_{3i} = (e_{3i} - b_{31}y_{1i} - b_{32}y_{2i} / b_{33};$

$x_{3i} = y_{3i}; \quad x_{2i} = y_{2i} - c_{23}x_{3i}; \quad x_{1i} = y_{1i} - c_{12}x_{2i} - c_{13}x_{3i}.$

Полагаем $i = 1$.

$$y_{11} = 1; \; y_{21} = (0 - 3 \cdot 1)/(-6) = 1/2;$$

$$y_{31} = (0 - 4 \cdot 1 - (-10) \cdot (1/2)/(2/3) = 3/2;$$

$$x_{31} = 3/2; \; x_{21} = (1/2) - (-5/6) \cdot (3/2) = 7/4;$$

$$x_{11} = 1 - 2 \cdot (7/4) - (-1) \cdot (3/2) = -1.$$

Полагаем $i = 2$.

$$y_{12} = 0;\ y_{22} = (1 - 3 \cdot 0)/(-6) = -1/6;$$

$$y_{32} = (0 - 4 \cdot 0 - (-10) \cdot (-1/6))/(2/3) = -5/2;$$

$$x_{32} = -5/2;\ x_{22} = (-1/6) - (-5/6) \cdot (-5/2) = -9/4;$$

$$x_{12} = 0 - 2 \cdot (-9/4) - (-1) \cdot (-5/2) = 2.$$

Полагаем $i = 3$.

$$y_{13} = 0;\ y_{23} = (0 - 3 \cdot 0)/(-6) = 0;$$

$$y_{33} = (1 - 4 \cdot 0 - (-10) \cdot 0)/(2/3) = 3/2;$$

$$x_{33} = 3/2;\ x_{23} = 0 - (-5/6) \cdot (3/2) = 5/4;$$

$$x_{13} = 0 - 2 \cdot (5/4) - (-1) \cdot (3/2) = -1.$$

Таким образом, получим

$$A^{-1} = \begin{pmatrix} -1 & 2 & -1 \\ 7/4 & -9/4 & 5/4 \\ 3/2 & -5/2 & 3/2 \end{pmatrix}.$$

1.1.4 Практическая работа № 4

Работа № 4 выполняется после изучения глав «Приближение функций» и «Численное дифференцирование».

Пример решения типового варианта

Задача 1

Для таблично заданной функции построить интерполяционный полином Ньютона третьего порядка.

x	3.5	3.55	3.60	3.65	3.70
y	33	34.8	36.8	39.1	41.9

Составим таблицу разностей

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
3.50	33.0	1.8	0.2	0.1	0.1
3.55	34.8	2.0	0.3	0.2	
3.60	36.8	2.3	0.5		
3.65	39.1	2.8			
3.70	41.9				

$$q = \frac{x - 3.5}{0.05} = 20(x - 3.5);\ n = 3;\ y_0 = 33.0 \cdot$$

$$P_3(x) = 33.0 + 1.8q + 0.2\frac{q(q-1)}{2} +$$

$$+0.1\frac{q(q-1)(q-2)}{6}.$$

Остаточный член:

$$R_3(x) = 0.1 \cdot \frac{q(q-1)(q-2)(q-3)}{4!},$$

Задача 2

Аппроксимировать полиномом Лежандра 5-ой степени функцию:

$$f(x) = \sin x, \quad x \in [-\pi/2,\ \pi/2]$$

<u>Справка</u>
$\int x \sin x dx = \sin x - \cos x;$
$\int x^2 \sin \ \ x dx = \frac{1}{2}x - \frac{1}{2}\sin x \cos x;$
$\int x^3 \sin x dx = (3x^2 - 6)\sin x - (x^3 - 6x \ \ )\cos x;$
$\int x^4 \sin x dx = (5x^4 - 60x^2 + 120)\sin x -$
$-(x^5 - 20x^3 - 120x)\cos x$

Решение.

Функция $f(x) = \sin x$ нечетная на интервале $x \in [-\pi/2,\ \pi/2]$. Так как полиномы Лежандра определены на интервале $[-1,1]$, то необходимо перейти к интервалу $[-1,1]$.

Введем переменную $y = \frac{2}{\pi}x$. Функция $f(y\pi/2) = \sin(y\pi/2)$ будет <u>нечетной</u> на интервале $y \in [-1,1]$. Т.к. полиномы Лежандра $P_k(x)$ являются нечетными при k нечетном, то отличными от нуля будут только коэффициенты C_k при нечетном k, т.е. C_1, C_3, C_5, а коэффициенты C_0, C_2, C_4 равны нулю.

Вычислим коэффициенты C_1, C_3, C_5.

$$C_1 = \frac{2}{3}\int_{-1}^{1} P_1(y)\sin\left(\frac{\pi}{2}y\right)dy = \frac{12}{\pi^2} = 3\left(\frac{2}{\pi}\right)^2 = 1.215854,$$

$$C_3 = \frac{7}{2}\int_{-1}^{1} P_3(y)\sin\left(\frac{\pi}{2}y\right)dy = \frac{7}{2}\int_{-1}^{1}\frac{1}{2}[5y^3 - 3y]\sin\left(\frac{\pi}{2}y\right)dy =$$

$$= 42\left(\frac{2}{\pi}\right)^2 - 105\left(\frac{2}{\pi}\right)^4 = -0.224891,$$

$$C_5 = \frac{11}{2}\int_{-1}^{1} P_5(y)\sin\left(\frac{\pi}{2}y\right)dy =$$

$$= \frac{11}{2}\cdot\frac{1}{8}\int_{-1}^{1}[63y^5 - 70y^3 + 15y]\sin\left(\frac{\pi}{2}y\right)dy =$$

$$= \frac{11}{16}63\left(\frac{2}{\pi}\right)^6\int_{-\pi/2}^{\pi/2} x^5\sin x dx - \frac{11}{16}70\left(\frac{2}{\pi}\right)^4\int_{-\pi/2}^{\pi/2} x^3\sin x dx +$$

$$+\frac{11}{6}15\left(\frac{2}{\pi}\right)^2\int_{-\pi/2}^{\pi/2} x\sin x dx = 165\left(\frac{2}{\pi}\right)^2 - 4620\left(\frac{2}{\pi}\right)^4 +$$

$$+10395\left(\frac{2}{\pi}\right)^6 = 0.0091979.$$

$$Q_5(y) = C_1 y + C_3\frac{1}{2}[5y^3 - 3y] + C_5\frac{1}{8}[63y^5 - 70y^3 + 15y];$$

$y \in [-1;1]$.

или

$$Q_5(y) = (C_1 - 3C_3 + \frac{15}{8}C_5)y + (\frac{5}{2}C_3 - \frac{70}{8}C_5)y^3 + \frac{63}{8}C_5 y^5 .$$

Подставим значения коэффициентов C_1, C_3, C_5, получим

$$Q_5(y) = 1.570436\ y - 0.642709\ y^3 + 0.072433\ y^5$$

Ответ:

$\sin(y\pi/2) = Q_5(y) = 1.570436\ y - 0.642709\ y^3 + 0.072433\ y^5$. Вычислим погрешность:

$$S = \int_{-1}^{1} [\sin(\frac{\pi}{2}y) - Q_5(y)]^2 = \frac{2}{\pi}\int_{-\pi/2}^{\pi/2} \sin^2 x dx - \sum_k C_k \frac{2}{2k+1} =$$

$$= \frac{2\pi}{\pi 2} - (C_1^2/3 + C_3^2/7 + C_5^2/11) = 1 - 2 \cdot 0.4999 \approx 0$$

Задача 3

Пусть $f(x) = \sqrt{1-x^2}$, $x \in [-1,1]$. Необходимо аппроксимировать полиномом Чебышева 5-ой степени $P_5(x)$ и вычислить ошибку.

Решение. Т.к. $f(x)$ – четная, то отличными от нуля будут коэффициенты C_i с четными номерами, а с нечетными равны нулю.

По формуле (5.51) получим

$$C_0 = \frac{1}{\pi} x \Big|_{-1}^{+1} = \frac{2}{\pi};$$

$$C_2 = \frac{2}{\pi}\int_{-1}^{1}(x^2 - 1/2)dx = -\frac{2^3}{\pi}\frac{1}{3} = -\frac{8}{3\pi};$$

$$C_4 = \frac{2^7}{\pi}\int_{-1}^{1}(x^4 - x^2 + 1/8)dx = \frac{2^7}{\pi}[\frac{1}{5}x^5 - \frac{1}{3}x^3 + \frac{1}{8}x]\Big|_{-1}^{+1} =$$

$$= -\frac{2^7}{\pi}\frac{1}{60} = -\frac{32}{15\pi}.$$

В результате получим

$$\sqrt{1-x^2} \approx \frac{2}{\pi} - \frac{8}{3\pi}(x^2 - \frac{1}{2}) - \frac{32}{15\pi}(x^4 - x^2 + \frac{1}{8})$$

или $\sqrt{1-x^2} = \frac{46}{15\pi} - \frac{8}{15\pi}x^2 - \frac{32}{15\pi}x^4$.

Погрешность равна:

$$S = \int_{-1}^{1}\rho(x)f^2(x)dx - \sum_{k=0} C_k^2 \int_{-1}^{1}\rho(x)T_k^2(x)dx =$$

$$= \int_{-1}^{1}\sqrt{1-x^2}dx - [C_0^2\pi + C_2^2\frac{\pi}{8} + C_4^2\frac{\pi}{2^7}] =$$

$$= [\frac{1}{2}x\sqrt{1-x^2} + \frac{1}{2}\arcsin x]_{-1}^{+1} - [\frac{4}{\pi} + \frac{8}{9\pi} + \frac{8}{225\pi}] =$$

$$= \frac{\pi}{2} - 1.5675 = 0.0033.$$

Задача 4

Материальная точка M движется прямолинейно. Закон движения $S = f(t)$ представлен в виде таблицы.

Найти скорость v и ускорение w т. M в момент $t = t^*$ сек.

Решение.

- Найти $q = \frac{t^* - t_0}{h}$.
- Вычислить ΔS_0, $\Delta^2 S_0$, $\Delta^3 S_0$ в момент времени t_0.
- Построить полином Ньютона.

Скорость v - есть первая производная полинома Ньютона, ускорение w -вторая производная полинома Ньютона.

1.1.5 Практическая работа № 5

Работа № 5 выполняется после изучения глав «Численное интегрирование», «Численные методы решения обыкновенных дифференциальных уравнений» и «Интегральные уравнения».

Пример решения типового варианта

Задача 1. Вычислить интеграл

$$\int_{0.5}^{1} \frac{dx}{x} \text{ при } n = 5.$$

а) по формуле трапеций;
б) по формуле прямоугольников;
в) по формуле Симпсона;
г) по формуле Гаусса;
д) по формуле Чебышева.
Рассчитать погрешность.

Решение. Для 5 узлов интегрирования шаг сетки составит 0.125.

При решении будем пользоваться таблицей значений функции. Здесь $f(x) = 1/x$.

	x		$f(x)$
x_0	0.5	y_0	2
x_1	0.625	y_1	1.6
x_2	0.750	y_2	1.33
x_3	0.875	y_3	1.14
x_4	1.0	y_4	1

а) формула трапеций:

$$I=\frac{h}{2}\cdot(y_0+2(y_1+y_2+y_3)+y_4);$$

$$I=(0.125/2)\cdot(2+2(1.6+1.33+1.14)+1)=0.696.$$

$$R=\frac{h\cdot(b\text{-}a)}{12}M_2, \text{ где } M_2 = \max\left|f''(x)\right|;\ f'' = \frac{2}{x^3}.$$

Максимальное значение второй производной функции на интервале $[0.5, 1]$ равно: $\max\left\{\left|f''(x)\right|, x \in [0.5,1]\right\} = \dfrac{2}{(0.5)^3} = 16$,

поэтому $R = \dfrac{0.125 \cdot (1-0.5)}{12} \cdot 16 = 0.0833$;

б) Формула прямоугольников.

Для левосторонней формулы имеем $I = h \cdot (y_0 + y_1 + y_2 + y_3)$, отсюда $I = 0.125 \cdot (2+1.6+1.33+1.14) = 0.759$.

Погрешность $R = \dfrac{h \cdot (b-a)}{2} \cdot M_1$,

где $M_1 = \max\left\{\left|f'(x)\right|,\ x \in [0.5, 1]\right\}$; $f'(x) = -\dfrac{1}{x^2}$.

Отсюда $M_1 = \dfrac{1}{(0.5)^2} = 4$, $R = \dfrac{0.125 \cdot (1-0.5)}{2} \cdot 4 = 0.125$;

в) формула Симпсона:

$$I = \frac{2h}{6} \cdot \{y_0 + y_4 + 4(y_1 + y_3) + 2y_2\}.$$

$$I = \frac{2 \cdot 0.125}{6}\{2+1+4 \cdot (1.6+1.14)+2 \cdot 1.33\} = 0.693.$$

$R = \dfrac{h^4 \cdot (b-a)}{180} M_4$, где $M_4 = \max\left\{f^{(4)}(x),\ x \in [0.5, 1]\right\}$.

$$M_4 = \max\left\{\frac{24}{x^5},\ x \in [0.5,1]\right\} = 768;$$

$$R = \frac{(0.125)^4 \cdot (1-0.5)}{180} \cdot 768 = 5.2 \cdot 10^{-4}.$$

г) формула Гаусса:

$$I = \frac{(b-a)}{2} \cdot (A_1 \cdot f(x_1) + A_2 \cdot f(x_2) + A_3 \cdot f(x_3) + A_4 \cdot f(x_4) +$$
$$+ A_5 \cdot f(x_5)).$$

$x_i = \frac{1}{2}(b+a) + \frac{1}{2}(b-a) \cdot t_i$. Здесь A_i, t_i - табличные данные.

t_i, $(n=5)$		A_i $(n=5)$	
t_1	0.90617985	A_1	0.23692688
t_2	0.53846931	A_2	0.47862868
t_3	0	A_3	0.56888889
t_4	-0.53846931	A_4	0.47862868
t_5	-0.90617985	A_5	0.23692688

Вычисленные значения x_i и y_i приведены в таблице

x_1	0.9765	y_1	1.024
x_2	0.8846	y_2	1.130
x_3	0.75	y_3	1.333
x_4	0.6154	y_4	1.625
x_5	0.5234	y_5	1.911

$$I=\frac{(1\text{-}0.5)}{2} \cdot (0.2426+0.5408+0.7566+0.7777+0.4525)=0.6923.$$

д) формула Чебышева:

$$I = \int_a^b f(x)dx = \frac{b-a}{n} \cdot \sum_{i=1}^{n} f(x_i);$$

$x_i = \frac{1}{2}(b+a) + \frac{b-a}{2} t_i, \quad i = 1, 2, ..., n$ приведение интервала интегрирования к интервалу $[-1,1]$. Узлы t_i приведены в таблице для $n = 5$.

t_1	0.832498
t_2	0.374541
t_3	0
t_4	-0.374541

t_5	-0.832498

Вычислим значения аргумента x и значения функции $y = f(x)$ в узлах t_i:

x_1	0,958	y_1	1,043
x_2	0,844	y_2	1,185
x_3	0,75	y_3	1,333
x_4	0,656	y_4	1,524
x_5	0,542	y_5	1,845

Сумма значений функции равна 6,927.

$$I = \frac{(1-0.5)}{5} \cdot 6.927 = 0.6927 .$$

Задача 2

Решить дифференциальное уравнение

$$y' = y + 2x + 1, \ \ y(0) = 0, \ \ h = 0.5 .$$

Решение.

1) метод численного интегрирования:

Формулы, по которым подсчитывается значение очередного приближения для трех узлов ($x_0 = 0, \ x_1 = 0.5, \ x_2 = 1$):

$$y_n(x_1) - y(x_0) = \frac{h}{12}[5f(x_0, y_{n-1}(x_0)) + 8f(x_1, y_{n-1}(x_1)) - \\ - f(x_2, y_{n-1}(x_2))];$$

$$y_n(x_2) - y(x_0) = \frac{h}{3}[f(x_0, y_{n-1}(x_0)) + 4f(x_1, y_{n-1}(x_1)) + \\ + f(x_2, y_{n-1}(x_2))].$$

По условию, $y_0(x_i) = y(0) = 0$, $x = (0, \ 0.5, \ 1)$.

n	$y_n(x)$
1	$y_1(x_1) = \frac{0.5}{12} \cdot [5 \cdot 1 + 8 \cdot (2 \cdot 0.5 + 1) - (2 \cdot 1 + 1)] = \frac{18}{24} = \frac{3}{4}$

	$y_1(x_2)=\frac{0.5}{3}\cdot[1\cdot 1+4\cdot(2\cdot 0.5+1)+(2\cdot 1+1)]=2$
2	$y_2(x_1)=\frac{0.5}{12}\cdot[5\cdot 1+8\cdot(\frac{3}{4}+2\cdot 0.5+1)-(2+2\cdot 1+1)]=\frac{11}{12}$ $y_2(x_2)=\frac{0.5}{3}\cdot[1\cdot 1+4\cdot(\frac{3}{4}+2\cdot 0.5+1)+(2+2\cdot 1+1)]=\frac{17}{6}$
3	$y_3(x_1)=\frac{0.5}{12}\cdot[5\cdot 1+8\cdot(\frac{11}{12}+2\cdot 0.5+1)-(\frac{17}{6}+2\cdot 1+1)]=\frac{127}{144}\approx$ ≈ 0.882 $y_3(x_2)=\frac{0.5}{3}\cdot[1\cdot 1+4\cdot(\frac{11}{12}+2\cdot 0.5+1)+(\frac{17}{6}+2\cdot 1+1)]=\frac{107}{36}\approx$ ≈ 2.972

2) метод Рунге- Кутта
первого порядка

$$y_{i+1}=y_i+k1,\ \ k1=h\cdot f(x_i,y_i)$$

Результаты вычислений:

i	x_i	y_i	y_{i+1}
0	0	$y_0=0$	$y_1=0+0.5\cdot(0+2\cdot 0+1)=0.5$
1	0.5	$y_1=0.5$	$y_2=0.5+0.5\cdot(0.5+2\cdot 0.5+1)=1.75$
2	1	$y_2=1.75$	

3) второго порядка:

$$y_{i+1}=y_i+\frac{1}{2}\cdot[k1+k2];\quad k1=h\cdot f(x_i,y_i);\quad k2=h\cdot f(x_i+h,y_i+k1);$$

Результаты вычислений:

i	x_i	y_i	k1	k2	y_{i+1}
0	0	$y_0=0$	0.5	1.25	$y_1=0.875$
1	0.5	$y_1=0.875$	1.4375	2.65625	$y_2=2.9219$
2	1	$y_2=2.9219$			

1.2 Варианты индивидуальных заданий для выполнения практических работ.

Вариант№ 1

1. Найти корни уравнения

$$y = \sqrt{-x} + \frac{1}{\sqrt{4+x}}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 7x_1 + 2x_2 + 3x_3 = 15 \\ 5x_1 - 3x_2 + 2x_3 = 15 \\ 10x_1 - 11x_2 + 5x_3 = 36 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 2 & 2 \\ 1 & 1 & -1 & 3 \\ 1 & 1 & 1 & -1 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 2 & -1 \\ 3 & 0 & 2 \\ 4 & -2 & 5 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_2^3 \frac{dx}{\sqrt{x-2} \cdot \sqrt[4]{9-x^2}}$$

6. Вычислить неопределённый интеграл.

$$\int \frac{1}{\sqrt{x}} \cos \sqrt{x} dx$$

7. Вычислить производную функции.

$$y = 6x^{\frac{7}{2}} + 4x^{\frac{5}{2}} + 2x$$

8. Решить дифференциальное уравнение.

$$(4e^{3y} - x)dy = dx$$

9. Найти значения функции $f(x) = \sin x + 2x$

$$x = \begin{bmatrix} 0 \\ 0{,}1\pi \\ 0{,}2\pi \\ \\ 2\pi \end{bmatrix}$$

Вариант№ 2

1. Найти корни уравнения
$y = 5\sin x \cdot \sin 3x$.
2. Решить систему линейных уравнений.
$$\begin{cases} 3x_1 + x_2 - x_3 = -8 \\ 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \end{cases}$$
3. Найти определитель матрицы.
$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$
4. Вычислить обратную матрицу.
$$A = \begin{vmatrix} 3 & -4 & 5 \\ 2 & -3 & 1 \\ 3 & -5 & -1 \end{vmatrix}$$
5. Вычислить определённый интеграл.
$$\int_2^4 \frac{dx}{\sqrt{x+2} \cdot \sqrt[5]{9 - x^2}}$$
6. Вычислить неопределённый интеграл.
$$\int \frac{dx}{x(3 + \ln^2 x)}$$
7. Вычислить производную функции.
$y = \sqrt[3]{x^2} - 2\sqrt{x} + 5$
8. Решить дифференциальное уравнение.
$y' + 2y = 4x$

9. Найти значения функции $f(x) = \cos(x^2) + x$

$x = 0..3\pi, h = 0.2$

Вариант№ 3

1. Найти корни уравнения

$y = \sin x \cdot \sin 3x$.

2. Решить систему линейных уравнений.

$$\begin{cases} 2x_1 + 2x_2 - x_3 + x_4 = 4 \\ 4x_1 + 3x_2 - x_3 + 2x_4 = 6 \\ 8x_1 + 5x_2 - 3x_3 + 4x_4 = 12 \\ 3x_1 + 3x_2 - 2x_3 + 2x_4 = 6 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 7 & 1 \end{vmatrix}$$

4. Вычислить обратную матрицу.

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

5. Вычислить определённый интеграл.

$$\int_0^1 x^3 e^{x^2} dx$$

6. Вычислить неопределённый интеграл.

$$\int \cos^4 x \cdot \sin^3 x \cdot dx$$

7. Вычислить производную функции.

$$y = e^{x^x}$$

8. Решить дифференциальное уравнение.

$y' + 5y = e^{7x}$

9. Найти значения функции $f(x) = \ln(x^2 + 5x + 3)$

$x = -10..10, h = 0.5$

Вариант№ 4

1. Найти корни уравнения

$$y = \frac{3x^2 - 1}{x^2 + 5}$$

2. Решить систему линейных уравнений.

$$\begin{cases} x_1 + x_2 - 2x_3 = 6 \\ 2x_1 + 3x_2 - 7x_3 = 16 \\ 5x_1 + 2x_2 + x_3 = 16 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 3 & 3 & -4 & -3 \\ 0 & 6 & 1 & 1 \\ 5 & 4 & 2 & 1 \\ 2 & 3 & 3 & 2 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 3 & 3 & -4 & -3 \\ 0 & 6 & 1 & 1 \\ 5 & 4 & 2 & 1 \\ 2 & 3 & 3 & 2 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^4 \frac{xdx}{1 + \sqrt{x}}$$

6. Вычислить неопределённый интеграл.

$$\int \frac{\sqrt{x}}{x - \sqrt[3]{x^2}} dx$$

7. Вычислить производную функции.

$$y = x^x$$

8. Решить дифференциальное уравнение.

$$y' + 2xy = 6x$$

9. Найти значения функции $f(x) = tgx \cdot ctg(x^2 + 1)$

$$x = -\pi..\pi, h = 0.1\pi$$

Вариант№ 5

1. Найти корни уравнения
$y = \sin x + \sin 2x$.

2. Решить систему линейных уравнений.

$$\begin{cases} 7x_1 + 2x_2 + 3x_3 = 15 \\ 5x_1 - 3x_2 + 2x_3 = 15 \\ 10x_1 - 11x_2 + 5x_3 = 36 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 2 & 1 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 & 1 \\ 1 & 1 & 4 & 1 & 1 \\ 1 & 1 & 1 & 5 & 1 \\ 1 & 1 & 1 & 1 & 6 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 3 & 2 & 1 \\ 4 & 5 & 2 \\ 2 & 1 & 4 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_3^8 \frac{dx}{2 + \sqrt{x+1}}$$

6. Вычислить неопределённый интеграл.

$$\int \frac{\sqrt[5]{(x+2)^3}}{\sqrt{x+2} - \sqrt[5]{(x+2)^8}} dx$$

7. Вычислить производную функции.

$$y = a^{tgnx}$$

8. Решить дифференциальное уравнение.

$$y' + 2y = 4x$$

9. Найти значения функции $f(x) = x^3 + \dfrac{x^2}{2x-1} + 2x$

$x = 100..1000, h = 50$

Вариант№ 6

1. Найти корни уравнения

$$y = \frac{1}{2x-1}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \\ 2x_1 + x_2 - x_3 = -5 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 5 & 6 & 0 & 0 & 0 \\ 1 & 5 & 6 & 0 & 0 \\ 0 & 1 & 5 & 6 & 0 \\ 0 & 0 & 1 & 5 & 6 \\ 0 & 0 & 0 & 1 & 5 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_1^2 \frac{dx}{\sqrt[5]{8-x^3}}$$

6. Вычислить неопределённый интеграл.

$$\int \frac{3x+3}{x^2+4x+20}\,dx$$

7. Вычислить производную функции.

$$y = x(2x-1)(3x+2)$$

8. Решить дифференциальное уравнение.

$$\sqrt{1-x^3} \cdot y' = x^2 \cdot \sqrt{1-y^2}$$

9. Найти значения функции $f(x) = \cos(2x^2+1)+3$

$$x = -\frac{3\pi}{2} .. \frac{\pi}{2}, h = 0.1\pi$$

Вариант№ 7

1. Найти корни уравнения

$$y = \frac{\cos 2x}{\sin x}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 7x_1 + 2x_2 + 3x_3 = 15 \\ 5x_1 - 8x_2 + x_3 = 19 \\ x_1 - 10x_2 + 15x_3 = 27 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ -1 & 0 & 3 & 4 \\ -1 & -2 & 0 & 4 \\ -1 & -2 & -3 & 4 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^{\frac{\pi}{2}} x \cdot \sin 6x dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{(x+5)^2}{x^2 + 25} dx$$

7. Вычислить производную функции.

$$y = \frac{x^3 + 1}{x^2 - x + 2}$$

8. Решить дифференциальное уравнение.

$$y' - 3y = 2e^{5x}$$

9. Найти значения функции $f(x) = \sin^2(2x + 3x^2)$

$x = 0..3\pi, h = 0.2\pi$

Вариант№ 8

1. Найти корни уравнения

$$y = \frac{x}{\cos \pi x}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 3x_1 + x_2 - x_3 = -8 \\ 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ -1 & 0 & 3 & 4 & 5 \\ -1 & -2 & 3 & 4 & 5 \\ -1 & -2 & -3 & 4 & 5 \\ -1 & -2 & -3 & -4 & 5 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^e \ln(x+1)dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{1}{\sin^4 x} dx$$

7. Вычислить производную функции.

$$y = x^{\sin x}$$

8. Решить дифференциальное уравнение.

$$y' + 2xy = 2xy^3$$

9. Найти значения функции $f(x) = e^{2x^2+5x+3}$

$x = 0..10, h = 0.2$

Вариант№ 9

1. Найти корни уравнения

$$y = x^3 - x^5$$

2. Решить систему линейных уравнений.

$$\begin{cases} 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \\ 2x_1 + x_2 - x_3 = -5 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 3 & 2 & 2 & 2 & 2 \\ 2 & 3 & 2 & 2 & 2 \\ 2 & 2 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 & 2 \\ 2 & 2 & 2 & 2 & 3 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^5 x \cdot e^{2x} dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{\cos^3 x}{1 + \sin^2 x} dx$$

7. Вычислить производную функции.

$$y = (\sin x)^x$$

8. Решить дифференциальное уравнение.

$$(4e^{3y} - x)dy = dx$$

9. Найти значения функции $f(x) = ctg^2(\frac{1}{x^2 + 1} + 5)$

$$x = -\frac{\pi}{2} .. \frac{\pi}{2}, h = 0.5\pi$$

Вариант№ 10

1. Найти корни уравнения

$$y = \cos^2 \frac{x}{2}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 7x_1 + 2x_2 + 3x_3 = 15 \\ 5x_1 - 3x_2 + 2x_3 = 15 \\ 10x_1 - 11x_2 + 5x_3 = 36 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 3 & 2 & 2 & 2 & 2 \\ 2 & 3 & 2 & 2 & 2 \\ 2 & 2 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 & 2 \\ 2 & 2 & 2 & 2 & 3 \end{vmatrix}$$

4. Вычислить обратную матрицу.

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

5. Вычислить определённый интеграл.

$$\int_1^2 \frac{dx}{\sqrt{4 - x^2} \cdot \sqrt[3]{x - 1}}$$

6. Вычислить неопределённый интеграл.

$$\int xe^{2x} dx$$

7. Вычислить производную функции.

$$y = \frac{a - x}{a + x}$$

8. Решить дифференциальное уравнение.

$$(1 + x^2)y' = x(y + 1)$$

9. Найти значения функции $f(x) = \dfrac{1}{x^3 + 2x^2 + x} + x^2$

$$x = 10..55, h = 2$$

Вариант№ 11

1. Найти корни уравнения

$$y=\sqrt{-x^2+5}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 7x_1+2x_2+3x_3=15 \\ 5x_1-8x_2+x_3=19 \\ x_1-10x_2+15x_3=27 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 6 & -5 & -4 & 1 \\ -3 & -6 & -3 & -3 \\ -1 & -4 & 1 & -8 \\ -3 & -4 & -4 & 2 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A=\begin{vmatrix} 1 & 0 & 8 & 0 \\ 1 & -3 & 9 & -4 \\ 2 & 6 & 1 & 7 \\ 1 & -2 & -6 & -4 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_1^2 \frac{dx}{\sqrt[7]{32-x^5}}$$

6. Вычислить неопределённый интеграл.

$$\int e^{(\ln^2 x+2)}\cdot\frac{\ln x}{x}dx$$

7. Вычислить производную функции.

$$y=\frac{(x+1)^3}{x^{\frac{3}{2}}}$$

8. Решить дифференциальное уравнение.

$$y'+2xy=6x$$

9. Найти значения функции $f(x)=3x^5+3x^4+8x^3+2x^2+10$, $x=-100..100, h=10$

Вариант№ 12

1. Найти корни уравнения

$$y = \arcsin\frac{x-3}{x}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 3x_1 + x_2 - x_3 = -8 \\ 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 1 & 0 & 8 & 0 \\ 1 & -3 & 9 & -4 \\ 2 & 6 & 1 & 7 \\ 0 & -2 & -6 & -4 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 0 & 8 & 0 \\ 1 & -3 & 9 & -4 \\ 2 & 6 & 1 & 7 \\ 1 & -2 & -6 & -4 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_1^2 \frac{dx}{\sqrt[7]{32 - x^5}}$$

6. Вычислить неопределённый интеграл.

$$\int e^{(\ln^2 x + 2)} \cdot \frac{\ln x}{x} dx$$

7. Вычислить производную функции.

$$y = \frac{(x+1)^3}{x^{\frac{3}{2}}}$$

8. Решить дифференциальное уравнение.

$$y' + 2xy = 6x$$

9. Найти значения функции

$$f(x) = 3x^5 + 3x^4 + 8x^3 + 2x^2 + 10, \quad x = -100..100, h = 10$$

Вариант№ 13

1. Найти корни уравнения

$$y = \sqrt[4]{6x - x^2 - 5}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 7x_1 + 2x_2 + 3x_3 = -15 \\ 5x_1 - 8x_2 + x_3 = -19 \\ x_1 - 10x_2 + 15x_3 = -27 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 2 & 4 & 1 & 1 \\ -6 & 0 & -9 & 1 \\ 2 & 0 & 2 & -4 \\ 2 & -6 & -6 & -7 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} -4 & -1 & -1 & 6 \\ 4 & 8 & 8 & 0 \\ -2 & -1 & 8 & -2 \\ 6 & 1 & 8 & 0 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \sin^3 x \cdot \cos x dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{(4x+2)dx}{\sqrt{1-(x+1)^2}}$$

7. Вычислить производную функции.

$$y = \log_3(x^2 - \sin x)$$

8. Решить дифференциальное уравнение.

$$y^2 y' - x^2 y^3 = x^2$$

9. Найти значения функции

$$f(x) = 1 + ctgx \cdot \sin^2 x^2, \quad x = \pi .. \frac{3\pi}{2}, h = 0.1\pi$$

Вариант№ 14

1. Найти корни уравнения

$$y = \sqrt{-x} + \frac{1}{\sqrt{4+x}}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 7x_1 + 2x_2 + 3x_3 = 15 \\ 5x_1 - 3x_2 + 2x_3 = 15 \\ 10x_1 - 11x_2 + 5x_3 = 36 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} -4 & -1 & -1 & 6 \\ 4 & 8 & 8 & 0 \\ -2 & -1 & 8 & -2 \\ 6 & 1 & 8 & 0 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 2 & 4 & 1 & 1 \\ -6 & 0 & -9 & 1 \\ 2 & 0 & 2 & -4 \\ 2 & -6 & -6 & -7 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^1 e^{x^2} x dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{\sqrt[4]{x-1}+1}{\sqrt{x-1}+\sqrt[4]{(x-1)^3}} dx$$

7. Вычислить производную функции.

$$y = \frac{a}{2}(e^{\frac{x}{a}} - e^{-\frac{x}{a}})$$

8. Решить дифференциальное уравнение.

$$xdy + ydx = 0$$

9. Найти значения функции $f(x) = tg\,\frac{x^3+4+1}{x^2+2}$

$x = -50..10, h = 0.5$

Вариант№ 15

1. Найти корни уравнения

$$y = \ln\frac{1-x}{1+x}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \\ 2x_1 + x_2 - x_3 = -5 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 2 & 2 \\ 1 & 1 & -1 & 3 \\ 1 & 1 & 1 & -1 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 5 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^{\frac{\pi}{2}} \cos^4 x \cdot \sin^3 x dx$$

6. Вычислить неопределённый интеграл.

$$\int x \sin 5x dx$$

7. Вычислить производную функции.

$$y = \frac{x^3 + 1}{x^2 - x + 2}$$

8. Решить дифференциальное уравнение.

$$(x^2 - xy + y^2)dy + y^2 dx = 0$$

9. Найти значения функции $f(x) = \dfrac{1}{x^3 + 4x^4 + 5}$

$$x = 1..1000, h = 5.2$$

Вариант№ 16

1. Найти корни уравнения

$y = x^2 \cdot \cos x$

2. Решить систему линейных уравнений.

$$\begin{cases} x_1 + x_2 - 2x_3 = 6 \\ 2x_1 + 3x_2 - 7x_3 = 16 \\ 5x_1 + 2x_2 + x_3 = 16 \end{cases}$$

3. Найти определитель матрицы.

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

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 0 & 0 & 1 & -1 \\ 0 & 3 & 1 & 4 \\ 2 & 7 & 6 & -1 \\ 1 & 2 & 2 & -1 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_{64}^{4096} \frac{\sqrt{x}}{x - \sqrt[3]{x^2}}\, dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{(3x + 2)dx}{\sqrt{5 - 4x^2 + 8x}}$$

7. Вычислить производную функции.

$$y = \frac{x^p}{x^m - a^m}$$

8. Решить дифференциальное уравнение.

$xy' = y(\ln y - \ln x)$

9. Найти значения функции

$f(x) = ctg(2x^2\sqrt{x})$, $x = -\pi .. \frac{\pi}{2}, h = 0.1$

Вариант№ 17

1. Найти корни уравнения

$$y = \arccos\frac{x-4}{x}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 2x_2 + 6x_3 = -7 \\ 2x_1 + x_2 - x_3 = -5 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} -2 & 2 & 6 & 4 \\ -1 & 9 & -5 & 4 \\ 4 & 3 & -1 & 1 \\ 1 & 4 & -6 & -3 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} -4 & 2 & -3 & 8 \\ -1 & -1 & 0 & -2 \\ 0 & -2 & 0 & 1 \\ 7 & 1 & 1 & -5 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_1^2 \frac{dx}{\sqrt[4]{16-x^4}}$$

6. Вычислить неопределённый интеграл.

$$\int (1+2\sin x)^3 \cos x dx$$

7. Вычислить производную функции.

$$y = (1-4x^3)(1+2x^2)$$

8. Решить дифференциальное уравнение.

$$2xydx + x^2 dy = 0$$

9. Найти значения функции

$$f(x) = \sqrt{\sin^2(x+2)}\,,\ x = -\frac{\pi}{2}..3\pi, h = 0.1$$

Вариант№ 18

1. Найти корни уравнения

$$y = \frac{2^x + 2^{-x}}{2^x - 2^{-x}}$$

2. Решить систему линейных уравнений.

$$\begin{cases} 3x_1 + x_2 - x_3 = -8 \\ 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 3 & 4 & 5 \\ 1 & 2 & 3 \\ 8 & 7 & 1 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^{\frac{\pi}{3}} \frac{dx}{\cos^4 x}$$

6. Вычислить неопределённый интеграл.

$$\int arct5xdx$$

7. Вычислить производную функции.

$$y = \frac{t^3}{1 + t^2}$$

8. Решить дифференциальное уравнение.

$$(y - x)dy + ydx = 0$$

9. Найти значения функции $f(x) = \ln 2x$

$x = 1..100, h = 10$

Вариант№ 19

1. Найти корни уравнения

$y = \sqrt[3]{x - 1}$

2. Решить систему линейных уравнений.

$$\begin{cases} 3x_1 + x_2 - x_3 = -8 \\ 5x_1 + 8x_2 + x_3 = 2 \\ 3x_1 - 2x_2 + 6x_3 = -7 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} 3 & 3 & -4 & -3 \\ 0 & 6 & 1 & 1 \\ 5 & 4 & 2 & 1 \\ 2 & 3 & 3 & 2 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 1 & 2 & 2 \\ 2 & 1 & -2 \\ 2 & -2 & 1 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^{\frac{\pi}{2}} \frac{\sqrt[5]{\sin^2 x}}{x} dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{dx}{x^2 + 4x + 20}$$

7. Вычислить производную функции.

$$y = \frac{2x^4}{b^2 - x^2}$$

8. Решить дифференциальное уравнение.

$x^3 (y^2 - 1)dx - (1 + x^4)2ydy = 0$

9. Найти значения функции

$f(x) = \sin(x + 2) + \cos^2(x + 3)$, $x = -3\pi..3\pi, h = 0.5\pi$

Вариант№ 20

1. Найти корни уравнения

$y = \sqrt{1 - \cos^2 x}$

2. Решить систему линейных уравнений.

$$\begin{cases} x_1 + x_2 - 2x_3 = 6 \\ 2x_1 + 3x_2 - 7x_3 = 16 \\ 5x_1 + 2x_2 + x_3 = 16 \end{cases}$$

3. Найти определитель матрицы.

$$\begin{vmatrix} -4 & 2 & -3 & 8 \\ -1 & -1 & 0 & -2 \\ 0 & -2 & 0 & 1 \\ 7 & 1 & 1 & -5 \end{vmatrix}$$

4. Вычислить обратную матрицу.

$$A = \begin{vmatrix} 3 & 3 & -4 & -3 \\ 0 & 6 & 1 & 1 \\ 5 & 4 & 2 & 1 \\ 2 & 3 & 3 & 2 \end{vmatrix}$$

5. Вычислить определённый интеграл.

$$\int_0^1 \frac{\ln(1 + \sqrt[7]{x})}{\sqrt{x}} dx$$

6. Вычислить неопределённый интеграл.

$$\int \frac{dx}{\sqrt{18x - 9x^2 - 5}}$$

7. Вычислить производную функции.

$y = (2x - 1) \cdot (x^2 - 6x + 3)$

8. Решить дифференциальное уравнение.

$\sqrt{9 - x^4}\, y' + x^3 (y^2 + 4) = 0$

9. Найти значения функции

$f(x) = \dfrac{ctg 2x}{\sin^2 3x}, \; x = -\pi..\pi, h = 0.01\pi$

2. ЛАБОРАТОРНЫЕ РАБОТЫ

2.1. Общие требования

По курсу «Численные методы» необходимо выполнить девять лабораторных работ. Лабораторные работы выполняются в течении семестра.

2.2. Требования к содержанию отчета

Отчет должен состоять из титульного листа установленной формы (см. Приложение А), задания и основной части. Основная часть содержит, как правило, следующие разделы: краткая теория, алгоритмы программы, результаты счета и выводы. Выводы должны содержать анализ результатов работы программы и сравнительный анализ, если программа реализует несколько методов. При анализе результатов следует обращать внимание на такие параметры, как точность вычислений, время счета, трудоемкость метода, скорость сходимости и количество итераций, объем памяти. Алгоритмы допускается приводить в виде блок-схемы или словесного описания. Желательно приводить алгоритмы отдельно для каждого метода, реализованного в программе. Листинг программы в отчет не включается. Пример оформления отчета приведен в Приложении Б.

2.3. Требования к оформлению программы

Программа должна быть написана на языке Паскаль. Текст программы должен содержать строку комментария с указанием имени студента и номера выполняемой работы. Программа должна сопровождаться комментариями. В комментариях указывают, какие методы реализуют подпрограммы, и спецификации входных и выходных параметров подпрограмм.

Пример. Подпрограмма реализует метод дихотомии нахождения корня функции. Фрагмент программы:

```
Procedure  Dich (f: func; a, b, eps: real; var x: real);
{Подпрограмма метода дихотомии.}
```

```
{f-функция; a, b- границы отрезка; eps- точность; x- значение корня.}
begin
…
end;
```

Отдельные фрагменты программы так же должны сопровождаться комментариями таким образом, чтобы легко можно было увязать текст программы с приведенным в отчете алгоритмом. Весь вывод на экран должен быть организован с пояснениями.

Пример. Необходимо вывести на экран полученное в программе приближенное решение и точное решение.

```
…
writeln(‘Приближенное решение                    Точное решение’);
for i:=1 to n do
   writeln (x[i]:12:5,’             ‘,x_t[i]:12:5);
{В массиве x содержится приближенное решение, в массиве x_t-точное}
```

В первой строке фрагмента программы содержится пояснение, которое сопровождает вывод результатов на экран.

Для стандартизации представления программ необходимо воспользоваться шаблонами, приведенными в Приложении В. Шаблоны включают описание всех функций и процедур, которые должна реализовать программа.

2.4. Темы лабораторных работ

2.4.1 Лабораторная работа № 1

Тема работы – «Оценка корректности вычислительных задач и алгоритмов». Цель работы: формулирование концептуальной, содержательной и математической модели для задач различных разделов физики. Построение математической модели. Обоснование выбора методов решения задачи. Оценка корректности вычислительной задачи и выбранного алгоритма.

2.4.2 Лабораторная работа № 2

Тема работы – «Решение уравнений с одной переменной». Программу составить согласно заданию. Обязательны для реализации следующие методы: Ньютона, хорд, итераций.

2.4.3 Лабораторная работа № 3

Тема работы – «Решение задач линейной алгебры». Программу составить согласно заданию. Обязательны для реализации следующие методы решения систем линейных уравнений: Гаусса, простой итерации, Зейделя. Методы вычисления определителей и обратной матрицы - полностью.

2.4.4 Лабораторная работа № 4

Тема работы – «Вычисление собственных чисел и собственных векторов». Программу составить согласно заданию.

2.4.5 Лабораторная работа № 5

Тема работы – «Интерполирование и численное дифференцирование функций». Программу составить согласно заданию.

2.4.6 Лабораторная работа № 6

Тема работы – «Приближение сплайнами». Программу составить согласно заданию.

2.4.7 Лабораторная работа № 7

Тема работы – «Численное интегрирование функций». Программу составить согласно заданию. Обязательны для реализации - формулы трапеций, Симпсона, прямоугольников и квадратурная формула Гаусса.

2.4.8. Лабораторная работа № 8

Тема работы – «Решение обыкновенных дифференциальных уравнений». Программу составить согласно заданию.

Реализовать:

1) решение дифференциального уравнения первого порядка методом Рунге- Кутта первого и четвертого порядков точности с контролем погрешности на шаге;

2) решение системы дифференциальных уравнений первого порядка;

3) решение дифференциального уравнения n-го порядка.

3. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Зариковская Н.В. Численные методы. Учебное пособие. 2011 (в электронном виде)
2. Мицель А.А. Вычислительная математика. Учебное пособие. –Томск: ТМЦ ДО, 2001.– 228с. (18 экз)
3. Мицель А.А. Практикум по численным методам. – Томск: ТУСУР, 2004. –196 с. (18 экз)
4. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль. – Томск: МП ”Раско”, 1992. – 270с. (116 экз)
5. Романенко В.В. Вычислительная математика. Лабораторный практикум. –Томск: ТУСУР, 2006. –114с.
6. Амосов А.А., Дубинский Ю.А., Конченова Н.В. Вычислительные методы для инженеров. М.: Высшая школа,1994.
7. Бахвалов, Жидков Н.П., Кобельков Г.М. Численные методы. М.: Наука,1987.
8. Крылов В.И., Бобков В.В., Монастырский П.И. Вычислительные методы. М.: Наука, 1977. Т.т. 1,2.
9. Демидович Б.П., Марон И.А. Основы вычислительной математики. М.: Наука, 1979.
10. Демидович Б.П. и др. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения. М.: Физматгиз, 1963.
11. Зариковская Н.В. Информатика: учебное пособие. – Томск: ТУСУР. 2007. – 182 с.
12. Кирьянов Д.В. Высшая математика на MathCAD. Видеокурс. http://www.intuit.ru/department/mathematics/basemathcad/
13. Бояршинов Б.С. Численные методы. Видеокурс. http://www.intuit.ru/department/mathematics/nummeth/
14. Язык программирования FreePascal, Pascal ABC.

ПРИЛОЖЕНИЕ А

Форма титульного листа к отчету по лабораторной работе

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики
(ЭМИС)

ОТЧЕТ
По лабораторной работе №__
по дисциплине «Прикладная информатика»

Выполнил
Студент группы _____ П. В. Иванов
(дата выполнения работы)

Принял
Доцент кафедры ЭМИС
___________________ Н.В.Зариковская

2018

ПРИЛОЖЕНИЕ Б

Пример оформления отчета по лабораторной работе

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики (ЭМИС)

ОТЧЕТ
По лабораторной работе №__
по дисциплине «Прикладная информатика»

Выполнил
Студент группы _____ П. В. Иванов
(дата выполнения работы)

Принял
Доцент кафедры ФЭ
___________________ Н.В.Зариковская

2018

Задание

1. Написать программу отделения корней.
2. Написать программу поиска корней двумя методами: перебора и хорд.

Вариант: V=(32*21) div 100=6.

Исходные данные: $f(x) = \sqrt{4x+7} - 3\cos(x)$.

Краткая теория

Отделение корней.

Для нахождения приближённого значения корней с использованием ЭВМ поступают следующим образом. Задают сетку $\{x_i\}$: $a=x_1<x_2<\ldots<x_n=b$ и вычисляют значения функции $f(x_i)=f_i$. Если для двух соседних точек выполняется неравенство $f(x_i)*f(x_{i+!})<0$, то в интервале $[x_i,\ x_{i+1}]$ расположен по крайней мере один корень $\xi_m \in [x_i,x_{i+1}]$.

Далее, если положить $\xi_m=1/2(x_i+x_{i+1})$, то точность определения корня ξ_m равна $\Delta_m=1/2(x_{i+1}-x_i)$.

Далее надо убедиться, что на интервале $[x_i,x_{i+1}]$ существует только один корень.

Метод перебора.

Пусть на интервале [a,b] расположен один корень. Требуется найти ξ с точностью ε. Разобьём [a,b] на n равных частей

$$x_i=a+ih;\ h=(b-a)/n,\ i=1,\ldots,n,$$

где $n=(b-a)/\ \varepsilon$ и вычисляют значение функции $f_i=f(x_i)$.

Если для двух соседних точек выполняется неравенство $f_i*f_{i+!}<0$, то полагают $\xi=1/2(x_i+x_{i+1})$.

Погрешность составит $\Delta=h/2=\varepsilon/2$.

Метод хорд.

Если b - неподвижный конец, то $x_{n+1}=x_n-(f(x_n)/(f(b)-f(x_n))*$ $*(b-x_n)$.

Если $f(x_n)*f(b)<0$ или $f(\ x_n)*f(a)>0$, то неподвижен конец b, конец a сдвигается.

Если a - неподвижный конец, то $x_{n+1}=a-(f(a))/(f(x_n)-f(a))*(x_n-a)$.

Если $f(x_n)*f(b)>0$ или $f(x_n)*f(a)<0$, то неподвижен конец a, конец b сдвигается.

Останов: $|\ x_{n+1}-x_n\ |<\varepsilon$ или $f\ (x_n\ )<\varepsilon$.

Алгоритмы используемых методов

1. Схема алгоритма отделения корней. Сетка равномерная. (См. рисунок 1).

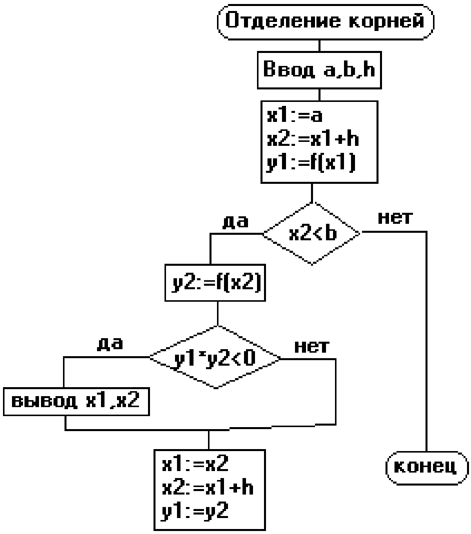

Рисунок 1 - Схема алгоритма отделения корней

2. Схема алгоритма метода перебора. (См. рисунок 2).

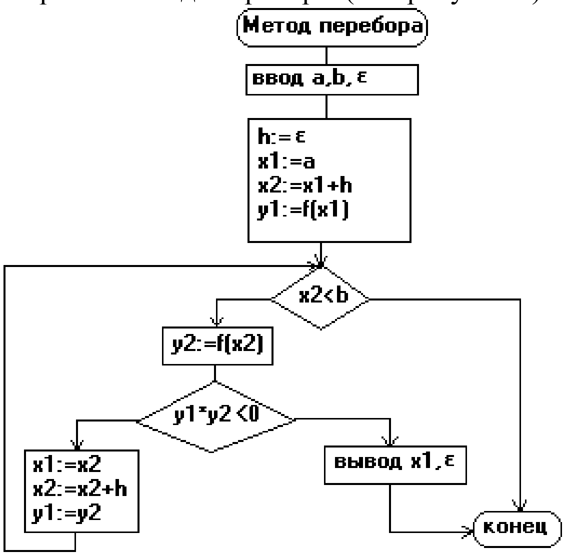

Рисунок 2 - Схема алгоритма метода перебора

3. Описание алгоритма метода хорд.
1. Ввод a,b,ε.
2. X:=a-f(a)*(b-a)/(f(b)-f(a)).
3.1. Если F(x)*F(b)<0 то a:=x.
Если F(x)*F(a)<0 то b:=x.
3.2. Пересчитать X по формуле п.2.
3.3. Выполнять п.3, пока abs(b-a)<=eps.
4. Вывод результата –x.

Результаты счета

Метод отделения корней
Введите границы a, b:
-10
10
Введите шаг h
5
На интервале от -10 до 10 обнаружен корень на интервале [0,5]
Метод перебора
Введите границы a, b, точность ε:
0
5
0.0001
Корень функции на интервале [0,5] x= 0,3015968.
Значение функции f(x)=301.513671875e-03.
Число итераций n=50000.
Метод хорд
Введите границы a, b, точность ε:
0
5
0.0001
Корень функции на интервале [0,5] x= 0,3015436.
Значение функции f(x)=301.513671764e-03.
Число итераций n=5.

Вывод

В результате выполнения работы реализованы метод отделения корней, метод перебора и метод хорд нахождения корней функции. Анализ результатов показывает, что метод хорд имеет более высокую скорость сходимости, чем метод перебора, а так же занимает меньше памяти при вычислениях. Количество узлов в методе перебора зависит от заданного числа шагов (заданной точности нахождения корня), а в методе хорд – от вида функции. Нахождение корня с любой, наперед заданной точностью возможно обоими методами, однако с увеличением точности метод перебора становится неэффективным с точки зрения использования памяти, времени счета и количества итераций.

ПРИЛОЖЕНИЕ В

Шаблоны для лабораторных работ

ЛАБОРАТОРНАЯ РАБОТА № 2

```
{Решение уравнений с одной переменной: шаблон программы}
{Программа написана для компиляторов Turbo Pascal/Borland
Pascal фирмы Borland inc.}
uses Crt;

Function Pow(A,B: Real):Real; {Вычисляет степень A^B}
 function relerror(observed, actual: real):real;
  begin
   if actual=0.0 then relerror:=abs(observed)
   else relerror:=abs(observed/actual-1);
  end;
var
 i:Byte;
 Res:Real;
begin
  if (Trunc(b)=b) and (b>0) then
   begin
    Res:=1;
    for i:=1 to Trunc(b) do
     Res:=Res*a;
    pow:=Res;
    Exit;
   end;
  if a=0 then
    if b=0 then pow:=1                  { 0^0 = 1 }
    else if b<0 then
    else pow:=0                         { 0^x = 0 }
  else if a<0 then
    if abs(b)<1e-10 then pow:=1
    else if relerror(b,round(b))<1e-8 then
      pow:=(1-2*ord(odd(round(b))))*exp(b*ln(abs(a)))
    else if (relerror(1/b,round(1/b))<1e-8) and odd(round(1/b)) then
```

```
      pow:=-exp(b*ln(abs(a)))
    else
  else pow:=exp(b*ln(a))
end;

Function F(N:Integer;X:Real):Real; (*Вычисляет значение функции из вариантов
                                     заданий с номером N в точке X          *)
  Begin
   case N of
    1: F:=0.008*X*X*X-cos(X);
    2: F:=X-10*sin(X);
    3: F:=Pow(2,-X)-Sin(X);
    4: F:=Pow(2,X)-2*Cos(X);
    5: F:=Ln(X+5)-Cos(X);
    6: F:=Sqrt(4*X+7)-3*Cos(X);
    7: F:=X*Sin(X)-1;
    8: F:=8*Cos(X)-X-6;
    9: F:=Sin(X)-0.2*X;
    10: F:=10*Cos(X)-0.1*X*X;
    11: F:=2*Ln(X+7)-5*Sin(X);
    12: F:=4*Cos(X)+0.3*X;
    13: F:=5*Sin(2*X)-Sqrt(1-X);
    14: F:=1.2*Pow(X,4)+2*X*X*X-24.1-13*X*X-14.2*X;
    15: F:=2*X*X-5-Pow(2,X);
    16: F:=0.5*Sqr(X)-10+Pow(2,-X);
    17: F:=4*Pow(X,4)-6.2-Cos(0.6*X);
    18: F:=3*Sin(8*X)-0.7*X+0.9;
    19: F:=1.2-Ln(X)-4*Cos(2*X);
    20: F:=Ln(X+6.1)-2*Sin(X-1.4);
    21: F:=Sqr(X-1)*Pow(X-3,3);
   end;
  End;

Function dF(N:Integer;X:Real):Real; (*Вычисляет значение первой производной
                                      функции с номером N в точке X         *)
  Begin
```

```
   case N of
    1: dF:=0.024*X*X+sin(X);
    2: dF:=1-10*cos(X);
    3..21: {остальные функции}
   end;
  End;

Function dF2(N:Integer;X:Real):Real; (*Вычисляет значение второй
производной
                                     функции с номером N в точке X         *)
  Begin
   case N of
    1: dF2:=0.048*X+cos(X);
    2: dF2:=10*sin(X);
    3..21: {остальные функции}
   end;
  End;

Procedure Dihotomy(Funct:Integer; A,B:Real; E:Real;
              var X, Fx:Real; var Iter:Integer; var alpha:Real);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для ре-
шения уравнений
   методом дихотомии.
   Входные параметры:
    Funct:Integer    -  номер уравнения;
    A,B:Real         -  интервал для поиска корней;
    E:Real           -  заданная точность.
   Выходные параметры, возвращаемые процедурой:
    X:Real           -  найденный  корень уравнения;
    Fx:Real          -  значение функции в заданном корне;
    Iter:Integer     -  количество итераций, за которое был найден ко-
рень;
    alpha:Real       -  параметр сходимости.*)
 End;

Procedure Newton(Funct:Integer; A,B:Real; E,E1:Real;
              var X, Fx:Real; var Iter:Integer; var alpha:Real);
```

```
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для ре-
шения уравнений
   методом Ньютона.
   Входные параметры:
    Funct:Integer    -  номер уравнения;
    A,B:Real         -  интервал для поиска корней;
    E,E1:Real        -  заданные точности.
   Выходные параметры, возвращаемые процедурой:
    X:Real           -  найденный  корень уравнения;
    Fx:Real          -  значение функции в заданном корне;
    Iter:Integer     -  количество итераций, за которое был найден ко-
рень;
    alpha:Real       -  параметр сходимости.*)
 End;

Procedure Hord(Funct:Integer; A,B:Real; E:Real;
              var X, Fx:Real; var Iter:Integer; var alpha:Real);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для ре-
шения уравнений
   методом хорд.
   Входные параметры:
    Funct:Integer    -  номер уравнения;
    A,B:Real         -  интервал для поиска корней;
    E:Real           -  заданная точность.
   Выходные параметры, возвращаемые процедурой:
    X:Real           -  найденный  корень уравнения;
    Fx:Real          -  значение функции в заданном корне;
    Iter:Integer     -  количество итераций, за которое был найден ко-
рень;
    alpha:Real       -  параметр сходимости.*)
 End;

Procedure Combin(Funct:Integer; A,B:Real; E,E1:Real;
              var X, Fx:Real; var Iter:Integer; var alpha:Real);
 Begin
```

```
 (*В теле этой процедуры напишите свою подпрограмму для решения уравнений
  комбинированным методом.
  Входные параметры:
   Funct:Integer   -  номер уравнения;
   A,B:Real        -  интервал для поиска корней;
   E,E1:Real       -  заданные точности.
  Выходные параметры, возвращаемые процедурой:
   X:Real          -  найденный  корень уравнения;
   Fx:Real         -  значение функции в заданном корне;
   Iter:Integer    -  количество итераций, за которое был найден корень;
   alpha:Real      -  параметр сходимости.*)
 End;

Procedure Golden(Funct:Integer; A,B:Real; E:Real;
           var X, Fx:Real; var Iter:Integer; var alpha:Real);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для решения уравнений
  методом золотого сечения.
  Входные параметры:
   Funct:Integer   -  номер уравнения;
   A,B:Real        -  интервал для поиска корней;
   E:Real          -  заданная точность.
  Выходные параметры, возвращаемые процедурой:
   X:Real          -  найденный  корень уравнения;
   Fx:Real         -  значение функции в заданном корне;
   Iter:Integer    -  количество итераций, за которое был найден корень;
   alpha:Real      -  параметр сходимости.*)
 End;

Procedure Iteration(Funct:Integer; A,B:Real; E:Real;
           var X, Fx:Real; var Iter:Integer; var alpha:Real);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для решения уравнений
```

```
  методом итераций.
  Входные параметры:
   Funct:Integer    -  номер уравнения;
   A,B:Real         -  интервал для поиска корней;
   E:Real           -  заданная точность.
  Выходные параметры, возвращаемые процедурой:
   X:Real           -  найденный  корень уравнения;
   Fx:Real          -  значение функции в заданном корне;
   Iter:Integer     -  количество итераций, за которое был найден ко-
рень;
   alpha:Real       -  параметр сходимости.*)
 End;

Procedure Lab1(Funct:Integer; A,B:Real; E,E1:Real; Method:Integer;
            var X, Fx:Real; var Iter:Integer; var alpha:Real);
 Begin
  case Method of
   1:Dihotomy(Funct,A,B,E,X,Fx,Iter,alpha);
   2:Newton(Funct,A,B,E,E1,X,Fx,Iter,alpha);
   3:Hord(Funct,A,B,E,X,Fx,Iter,alpha);
   4:Combin(Funct,A,B,E,E1,X,Fx,Iter,alpha);
   5:Golden(Funct,A,B,E,X,Fx,Iter,alpha);
   6:Iteration(Funct,A,B,E,X,Fx,Iter,alpha);
  end;
 End;

var
 (*Входные данные:      *)
 Funct:Integer;  {Вариант задания}
 A,B:Real;       {Интервал [a,b]}
 N:Integer;      {Количество узлов}
 E,E1:Real;      {Точность E и E1}
 Method:Integer; {Метод решения уравнения}
 (*Выходные данные:     *)
 A1,B1:Real;     {Интервалы, где есть корни}
 X:Real;         {Корни уравнения}
 Fx:Real;        {Значения функции в найденных корнях}
 Iter:Integer;   {Число итераций}
```

```
 alpha:Real;     {Параметр сходимости}
 (*Вспомогательные переменные*)
 i,Error:Integer;
 myFile:Text;
Begin
 if ParamCount<>0 then
  begin
   {Считываем параметры, заданные с помощью командной стро-
ки}
   Val(ParamStr(1),Funct,Error);
   Val(ParamStr(2),A,Error);
   Val(ParamStr(3),B,Error);
   Val(ParamStr(4),N,Error);
   Val(ParamStr(5),E,Error);
   Val(ParamStr(6),E1,Error);
   Val(ParamStr(7),Method,Error);
   {Создаем файл результатов}
   Assign(myFile,'results.txt');
   ReWrite(myFile);
  end
 else
  begin
   {Считываем параметры с клавиатуры}
   WriteLn;
   Write('Введите номер функции (1..21): '); ReadLn(Funct);
   Write('Введите нижнюю границу интервала: '); ReadLn(A);
   Write('Введите верхнюю границу интервала: '); ReadLn(B);
   Write('Введите количество узлов: '); ReadLn(N);
   Write('Введите точность E: '); ReadLn(E);
   Write('Введите точность E1: '); ReadLn(E1);
   Write('Введите номер метода вычислений (1 - дихотомии; 2 -
Ньютона;',
     '3 - хорд; 4 - комбинированный; 5 - золотого сечения; 6 - итера-
ций) :');
   ReadLn(Method);
   AssignCRT(myFile);
   ReWrite(myFile);
  end;
```

```
 {Отделяем корни}
 for i:=0 to N-1 do
  if F(Funct,A+i*(B-A)/N)*F(Funct,A+(i+1)*(B-A)/N)<0 then {в этом
промежутке
                                                  находится корень}
   begin
    A1:=A+i*(B-A)/N;
    B1:=A+(i+1)*(B-A)/N;
    Lab1(Funct,A1,B1,E,E1,Method,X,Fx,Iter,alpha); {решаем уравне-
ние заданным
                                            методом}
    {Сохраняем результаты в файле}
    WriteLn(myFile,'Функция:            ',Funct);
    WriteLn(myFile,'Метод:              ',Method);
    WriteLn(myFile,'A:                  ',A1);
    WriteLn(myFile,'B:                  ',B1);
    WriteLn(myFile,'Корень:             ',X);
    WriteLn(myFile,'Значение функции:     ',Fx);
    WriteLn(myFile,'Количество итераций:  ',Iter);
    WriteLn(myFile,'Параметр сходимости:  ',alpha);
    WriteLn(myFile);
    if ParamCount=0 then
     begin
      WriteLn('Нажмите <Ввод> для продолжения');
      ReadLn;
     end;
   end;
End.
```

ЛАБОРАТОРНАЯ РАБОТА № 3

```
{Решение задач линейной алгебры: шаблон программы}
{Программа написана для компиляторов Turbo Pascal/Borland
Pascal фирмы Borland inc.}
uses Crt;

type
 MatrType=array[1..5,1..6] of Real;      {Тип, описывающий матрицу
[5x6]}
```

```
 VectorType=array[1..5] of Real;       {Вектор из 5-ти элементов}
type
 TestType=record
  Matr:MatrType;
  VarNum:Byte;
 end;

const
 Tests1:array[1..13] of TestType=
  ((Matr:((7,2,3,15,0,0),
        (5,-3,2,15,0,0),
        (10,-11,5,36,0,0),         {1}
        (15,7,2,3,0,0),
        (15,7,2,3,0,0));VarNum:3),
   (Matr:((1,1,-2,6,0,0),
        (2,3,-7,16,0,0),
        (5,2,1,16,0,0),            {2}
        (15,7,2,3,0,0),
        (15,7,2,3,0,0));VarNum:3),
   (Matr:((2,2,-1,1,4,0),
        (4,3,-1,2,6,0),
        (8,5,-3,4,12,0),         {3}
        (3,3,-2,2,6,0),
        (15,7,2,3,0,0));VarNum:4),
   (Matr:((5,8,1,2,0,0),
        (3,-2,6,-7,0,0),
        (2,1,-1,-5,0,0),           {4}
        (15,7,2,3,0,0),
        (15,7,2,3,0,0));VarNum:3),
   (Matr:((3.2,5.4,4.2,2.2,2.6,0),
        (2.1,3.2,3.1,1.1,4.8,0),
        (1.2,0.4,-0.8,-0.8,3.6,0),         {5}
        (4.7,10.4,9.7,9.7,-8.4,0),
        (15,7,2,3,0,0));VarNum:4),
   (Matr:((3.2,5.4,4.2,2.2,11.4,0),
        (2.1,3.2,3.1,1.1,9.2,0),
        (1.2,0.4,-0.8,-0.8,0.4,0),         {6}
        (4.7,10.4,9.7,9.7,30.4,0),
```

```
       (15,7,2,3,0,0));VarNum:4),
   (Matr:((3,1,-1,-8,0,0),
       (5,8,1,2,0,0),
       (3,-2,6,-7,0,0),         {7}
       (15,7,2,3,0,0),
       (15,7,2,3,0,0));VarNum:3),
  (Matr:((7,2,3,15,0,0),
       (5,-8,1,19,0,0),
       (1,-10,15,27,0,0),            {8}
       (15,7,2,3,0,0),
       (15,7,2,3,0,0));VarNum:3),
  (Matr:((7,2,3,-15,0,0),
       (5,-8,1,-19,0,0),
       (1,-10,15,-27,0,0),            {9}
       (15,7,2,3,0,0),
       (15,7,2,3,0,0));VarNum:3),
  (Matr:((6.087,-3.913,7.547,1.734,3.21,0),
       (1.739,0.869,1.887,0.73,6.35,0),
       (2.174,-1.305,2.83,1.04,1.5,0),            {10}
       (4.5,-1.305,1.887,0.541,-1.27,0),
       (15,7,2,3,0,0));VarNum:4),
  (Matr:((2.67,5.1,3.31,5.64,4.76,6.19),
       (4.44,7.5,4.67,5.7,6.14,6.95),
       (5.33,9.8,8.67,4.8,7.33,12.2),         {11}
       (3.56,5.3,4.15,3.69,3.25,5.97),
       (1.78,4.17,2.67,4.69,3.75,4.42));VarNum:5),
  (Matr:((4.1,0.1,0.2,0.2,21.14,0),
       (0.3,5.3,0.9,-0.1,-17.82,0),
       (0.2,0.3,3.2,0.2,9.02,0),            {12}
       (0.1,0.1,0.2,-9.1,17.08,0),
       (15,7,2,3,0,0));VarNum:4),
  (Matr:((2.4,0.2,-0.3,-1.1,5.8,23.84),
       (0.3,0.1,1.1,10.2,1,38.85),
       (0.5,-6.2,0.1,1.5,-1.2,17.23),            {13}
       (0.1,2.1,5.1,0.2,-0.3,6.56),
       (2.5,0.1,0.2,0.3,0.4,6.63));VarNum:5));
Tests2Num:array[1..10] of Byte=(4,3,3,4,5,5,5,5,5,5);
Tests2:array[1..10] of MatrType=
```

```
(((1,1,1,1,1,0),
  (1,-1,2,2,1,0),
  (1,1,-1,3,3,0),
  (1,1,1,-1,-1,0),
  (1,1,1,1,1,0)),
 ((1,2,3,1,1,0),
  (4,5,6,2,1,0),
  (7,8,9,-1,3,0),
  (1,1,1,1,-1,0),
  (1,1,1,1,1,0)),
 ((3,4,5,1,1,0),
  (1,2,3,2,1,0),
  (8,7,1,-1,3,0),
  (1,1,1,1,-1,0),
  (1,1,1,1,1,0)),
 ((3,3,-4,-3,1,0),
  (0,6,1,1,1,0),
  (5,4,2,1,3,0),
  (2,3,3,2,-1,0),
  (1,1,1,1,1,0)),
 ((2,1,1,1,1,0),
  (1,3,1,1,1,0),
  (1,1,4,1,1,0),
  (1,1,1,5,1,0),
  (1,1,1,1,6,0)),
 ((5,6,0,0,0,0),
  (1,5,6,0,0,0),
  (0,1,5,6,0,0),
  (0,0,1,5,6,0),
  (0,0,0,1,5,0)),
 ((1,2,3,4,5,0),
  (-1,0,3,4,5,0),
  (-1,-2,0,4,5,0),
  (-1,-2,-3,0,5,0),
  (-1,-2,-3,-4,0,0)),
 ((3,2,2,2,2,0),
  (2,3,2,2,2,0),
  (2,2,3,2,2,0),
```

```
   (2,2,2,3,2,0),
   (2,2,2,2,3,0)),
  ((1,1,1,1,1,0),
   (1,2,2,2,2,0),
   (1,2,3,3,3,0),
   (1,2,3,4,4,0),
   (1,2,3,4,5,0)),
  ((1,2,3,4,5,0),
   (2,2,3,4,5,0),
   (3,3,3,4,5,0),
   (4,4,4,4,5,0),
   (5,5,5,5,5,0)));
Tests3Num:array[1..14] of Byte=(3,3,4,3,4,3,5,4,3,3,4,5,4,5);
Tests3:array[1..14] of MatrType=
 (((1,2,-1,1,1,0),
   (3,0,2,2,1,0),
   (4,-2,5,-1,3,0),
   (1,1,1,1,-1,0),
   (1,1,1,1,1,0)),
  ((3,-4,5,1,1,0),
   (2,-3,1,2,1,0),
   (3,-5,-1,-1,3,0),
   (1,1,1,1,-1,0),
   (1,1,1,1,1,0)),
  ((3,3,-4,-3,1,0),
   (0,6,1,1,1,0),
   (5,4,2,1,3,0),
   (2,3,3,2,-1,0),
   (1,1,1,1,1,0)),
  ((3,2,1,-3,1,0),
   (4,5,2,1,1,0),
   (2,1,4,1,3,0),
   (2,3,3,2,-1,0),
   (1,1,1,1,1,0)),
  ((1,1,1,1,1,0),
   (1,1,-1,-1,1,0),
   (1,-1,1,-1,1,0),
   (1,-1,-1,1,1,0),
```

(1,1,1,1,6,0)),
((2,5,7,0,0,0),
(6,3,4,0,0,0),
(5,-2,-3,6,0,0),
(0,0,1,5,6,0),
(0,0,0,1,5,0)),
((1,1,1,1,1,0),
(0,1,1,1,1,0),
(0,0,1,1,1,0),
(0,0,0,1,1,0),
(0,0,0,0,1,0)),
((1,1,1,1,2,0),
(1,1,-1,-1,2,0),
(1,-1,0,0,2,0),
(0,0,1,-1,2,0),
(2,2,2,2,3,0)),
((2,7,3,1,1,0),
(3,9,4,2,2,0),
(1,5,3,3,3,0),
(1,2,3,4,4,0),
(1,2,3,4,5,0)),
((1,2,2,0,0,0),
(2,1,-2,0,0,0),
(2,-2,1,6,0,0),
(0,0,1,5,6,0),
(0,0,0,1,5,0)),
((3,3,-4,-3,5,0),
(0,6,1,1,5,0),
(5,4,2,1,5,0),
(2,3,3,2,5,0),
(-1,-2,-3,-4,0,0)),
((1,1,0,0,0,0),
(0,1,1,0,0,0),
(0,0,1,1,0,0),
(0,0,0,1,1,0),
(0,0,0,0,1,0)),
((0,0,1,-1,1,0),
(0,3,1,4,2,0),

```
   (2,7,6,-1,3,0),
   (1,2,2,-1,4,0),
   (1,2,3,4,5,0)),
  ((1,0,0,0,1,0),
   (0,2,0,0,0,0),
   (0,0,3,0,0,0),
   (0,0,0,4,0,0),
   (0,0,0,0,5,0)));
Procedure Gauss1(Matr:MatrType; N:Integer; var X,Nev:VectorType);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для реше-
ния СЛАУ
  методом Гаусса.
  Входные параметры:
   Matr:MatrType    -  матрица [NxN+1], состоящая из коэффициен-
тов СЛАУ,
                    последий столбец ([i,N+1]) - правая часть
                    i-го уравнения;
   N:Integer       -  размерность матрицы Matr.
  Выходные параметры, возвращаемые процедурой:
   X:VectorType     -  массив [1..N] решений системы (X1..XN);
   Nev:VectorType   -  массив [1..N] невязок системы;          *)
 End;

Procedure Ortogon1(Matr:MatrType; N:Integer; var
X,Nev:VectorType);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для реше-
ния СЛАУ
  методом ортогонолизации.
  Входные параметры:
   Matr:MatrType    -  матрица [NxN+1], состоящая из коэффициен-
тов СЛАУ,
                    последий столбец ([i,N+1]) - правая часть
                    i-го уравнения;
   N:Integer       -  размерность матрицы Matr.
  Выходные параметры, возвращаемые процедурой:
   X:VectorType     -  массив [1..N] решений системы (X1..XN);
```

```
   Nev:VectorType   -  массив [1..N] невязок системы;          *)
 End;

Procedure Decompos1(Matr:MatrType; N:Integer; var
X,Nev:VectorType);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для реше-
ния СЛАУ
  методом декомпозиции.
  Входные параметры:
   Matr:MatrType    -  матрица [NxN+1], состоящая из коэффициен-
тов СЛАУ,
                  последий столбец ([i,N+1]) - правая часть
                  i-го уравнения;
   N:Integer       -  размерность матрицы Matr.
  Выходные параметры, возвращаемые процедурой:
   X:VectorType     -  массив [1..N] решений системы (X1..XN);
   Nev:VectorType   -  массив [1..N] невязок системы;          *)
 End;

Procedure Iteration1(Matr:MatrType; N:Integer; E:Real; var
X,Nev:VectorType);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для реше-
ния СЛАУ
  методом простой итерации.
  Входные параметры:
   Matr:MatrType    -  матрица [NxN+1], состоящая из коэффициен-
тов СЛАУ,
                  последий столбец ([i,N+1]) - правая часть
                  i-го уравнения;
   N:Integer       -  размерность матрицы Matr;
   E:Real         -  точность вычислений.
  Выходные параметры, возвращаемые процедурой:
   X:VectorType     -  массив [1..N] решений системы (X1..XN);
   Nev:VectorType   -  массив [1..N] невязок системы;          *)
 End;
```

```
Procedure Zeidel1(Matr:MatrType; N:Integer; E:Real; var X,Nev:VectorType);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для решения СЛАУ
   методом Зейделя.
   Входные параметры:
    Matr:MatrType    -  матрица [NxN+1], состоящая из коэффициентов СЛАУ,
                        последий столбец ([i,N+1]) - правая часть
                        i-го уравнения;
    N:Integer        -  размерность матрицы Matr;
    E:Real           -  точность вычислений.
   Выходные параметры, возвращаемые процедурой:
    X:VectorType     -  массив [1..N] решений системы (X1..XN);
    Nev:VectorType   -  массив [1..N] невязок системы;          *)
 End;

Procedure Gauss2(Matr:MatrType; N:Integer; var Opr:Real);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для вычисления
   определителя матрицы методом Гаусса.
   Входные параметры:
    Matr:MatrType    -  квадратная матрица [NxN];
    N:Integer        -  размерность матрицы Matr.
   Выходные параметры, возвращаемые процедурой:
    Opr:Real         -  определитель матрицы Matr.     *)
 End;

Procedure Decompos2(Matr:MatrType; N:Integer; var Opr:Real);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для вычисления
   определителя матрицы методом декомпозиции.
   Входные параметры:
    Matr:MatrType    -  квадратная матрица [NxN];
    N:Integer        -  размерность матрицы Matr.
```

```
  Выходные параметры, возвращаемые процедурой:
   Opr:Real        -  определитель матрицы Matr.     *)
 End;

Procedure ObrMatr(Matr:MatrType; N:Integer; E:Real; Method:Integer;
                var ResMatr:MatrType);
 Begin
 (*В теле этой процедуры напишите свою подпрограмму для
нахождения обратной
  матрицы методами Гаусса, ортогонализации, декомпозиции, про-
стой итерации,
  Зейделя. Воспользуйтесь для этого соответствующими процеду-
рами для решения
  СЛАУ.
  Входные параметры:
   Matr:MatrType    -  квадратная матрица [NxN];
   N:Integer        -  размерность матрицы Matr;
   E:Real           -  точность вычислений (для итерационных мето-
дов).
   Method:Integer   -  номер метода вычислений:
                  Method == 1  -  метод Гаусса;
                  Method == 2  -  метод ортогонолизации;
                  Method == 3  -  метод декомпозиции;
                  Method == 4  -  метод простой итерации;
                  Method == 5  -  метод Зейделя.
  Выходные параметры, возвращаемые процедурой:
   ResMatr:MatrType    -     вычисленная   обратная   матрица.
*)
 End;

Procedure Lab2(WorkType,Funct,Method:Integer; E:Real;
          var   X,Nev:VectorType;   var   ResMatr:MatrType;   var
Opr:Real);
 var
  Matr:MatrType;
  N:Integer;
 Begin
  case WorkType of
```

```
   1:begin      {решение СЛАУ}
    Matr:=Tests1[Funct].Matr;
    N:=Tests1[Funct].VarNum;
    case Method of
     1:Gauss1(Matr,N,X,Nev);
     2:Ortogon1(Matr,N,X,Nev);
     3:Decompos1(Matr,N,X,Nev);
     4:Iteration1(Matr,N,E,X,Nev);
     5:Zeidel1(Matr,N,E,X,Nev);
    end;
   end;
   2:begin
    Matr:=Tests2[Funct];
    N:=Tests2Num[Funct];
    case Method of
     1:Gauss2(Matr,N,Opr);
     2:Decompos2(Matr,N,Opr);
    end;
   end;
   3:begin
    Matr:=Tests3[Funct];
    N:=Tests3Num[Funct];
    ObrMatr(Matr,N,E,Method,ResMatr);
   end;
  end;
 End;

var
 (*Входные данные:      *)
 WorkType:Integer; {Тип задания: 1 - решение СЛАУ; 2 - определи-
тели; 3 - обратные матрицы}
 Funct:Integer;    {Вариант задания}
 Method:Integer;   {Метод вычисления}
 E:Real;           {Точность (для итерационных методов)}
 (*Выходные данные:     *)
 X:VectorType;     {Вектор решения X}
 Nev:VectorType;   {Вектор невязки}
 ResMatr:MatrType; {Обратная матрица}
```

```
 Opr:Real;        {Определитель}
 (*Вспомогательные переменные*)
 N:Integer;       {Размерность матрицы}
 i,j,Error:Integer;
 myFile:Text;
Begin
 if ParamCount<>0 then
  begin
   {Считываем параметры, заданные с помощью командной стро-
ки}
   Val(ParamStr(1),WorkType,Error);
   Val(ParamStr(2),Funct,Error);
   Val(ParamStr(3),Method,Error);
   Val(ParamStr(3),E,Error);
   {Создаем файл результатов}
   Assign(myFile,'results.txt');
   ReWrite(myFile);
  end
 else
  begin
   {Считываем параметры с клавиатуры}
   WriteLn;
   Write('Введите тип задания (1 - решение СЛАУ; 2 - вычисление
определителя;',
        '3 - нахождение обратной матрицы): '); ReadLn(WorkType);
   case WorkType of
    1:begin       {решение СЛАУ}
     Write('Введите номер задания (1..13): '); ReadLn(Funct);
     Write('Введите метод решения СЛАУ (1 - Гаусса; 2 - ортогоно-
лизации;',
      '3 - декомпозиции; 4 - простой итерации; 5 - Зейделя): ');
ReadLn(Method);
     if Method>3 then
      begin
       Write('Введите точность вычислений: ');
       ReadLn(E);
      end;
    end;
```

```
   2:begin       {определители}
    Write('Введите номер задания (1..8): '); ReadLn(Funct);
    Write('Введите метод вычисления определителя (1 - Гаусса; 2 - декомпозиции): ');
    ReadLn(Method);
   end;
   3:begin       {обратные матрицы}
    Write('Введите номер задания (1..14): '); ReadLn(Funct);
    Write('Введите метод нахождения обратной матрицы (1 - Гаусса; 2 - ортогонолизации;',
     '3 - декомпозиции; 4 - простой итерации; 5 - Зейделя): '); ReadLn(Method);
    if Method>3 then
     begin
      Write('Введите точность вычислений: ');
      ReadLn(E);
     end;
   end;
  end;
  AssignCRT(myFile);
  ReWrite(myFile);
 end;
 if WorkType=1 then
  N:=Tests1[Funct].VarNum
 else
  N:=Tests3Num[Funct];
 Lab2(WorkType,Funct,Method,E,X,Nev,ResMatr,Opr);
 {Вывод результатов}
 WriteLn(myFile,'Тип задания:          ',WorkType);
 WriteLn(myFile,'Матрица:              ',Funct);
 case WorkType of
  1:for i:=1 to N do
    begin
     WriteLn(myFile,'X[',i,']:   ',X[i]);
     WriteLn(myFile,'Nev[',i,']: ',Nev[i]);
    end;
  2:WriteLn(myFile,'Определитель:          ',Opr);
  3:begin
```

```
    WriteLn(myFile,'Обратная матрица:');
    for i:=1 to N do
     begin
      for j:=1 to N do
       Write(myFile,ResMatr[i,j],' ');
      WriteLn(myFile);
     end;
   end;
  end;
End.
```

ЛАБОРАТОРНАЯ РАБОТА № 4

```
{Вычисление собственных чисел и собственных векторов: шаблон
программы}
{Программа написана для компиляторов Turbo Pascal/Borland
Pascal фирмы Borland inc.}
uses Crt;

type
 MatrType=array[1..4,1..4] of Real;     {Тип, описывающий матрицу
[4x4]}
 VectorType=array[1..4] of Real;     {Вектор из 4-х элементов}

const
 TestMatr:array[1..11] of MatrType=
  (((1,2,3,4),        {1}
    (2,1,2,3),
    (3,2,1,2),
    (4,3,2,1)),
   ((1,-3,3,0),       {2}
    (-2,-6,13,0),
    (-1,-4,8,0),
    (0,0,0,0)),
   ((2,-1,2,0),       {3}
    (5,-3,3,0),
    (-1,0,-2,0),
    (0,0,0,0)),
   ((0,1,0,0),        {4}
```

```
 (-4,4,0,0),
 (-2,1,2,0),
 (0,0,0,0)),
((4,-5,2,0),        {5}
 (5,-7,3,0),
 (6,-9,4,0),
 (0,0,0,0)),
((1,-3,4,0),        {6}
 (4,-7,8,0),
 (6,-7,7,0),
 (0,0,0,0)),
((7,-12,6,0),       {7}
 (10,-19,10,0),
 (12,-24,13,0),
 (0,0,0,0)),
((1,0,0,0),         {8}
 (1,2,1,0),
 (-1,0,1,0),
 (0,0,0,0)),
((2,6,-15,0),       {9}
 (1,1,-5,0),
 (1,2,-6,0),
 (0,0,0,0)),
((1,-3,1,0),        {10}
 (3,-3,-1,0),
 (3,-5,1,0),
 (0,0,0,0)),
((0,1,0,0),         {11}
 (1,1,-2,0),
 (1,-1,0,0),
 (0,0,0,0)));

Procedure Lab3(Matr:MatrType; N:Integer; var L:VectorType; var X:MatrType);
 (*В теле этой процедуры напишите свою подпрограмму для нахождения
  собственных чисел и собственных векторов.
  Входные параметры:
```

```
   Matr:MatrType     -  квадратная матрица, для которой находятся
собственные значения;
   N:Integer       -  размерность матрицы Matr.
  Выходные параметры, возвращаемые процедурой:
   L:VectorType    -  массив [1..N] собственных чисел;
   X:MatrType      -  двумерный массив, строки которого содержат
                    собственные векторы                    *)
 Begin

 End;

var
 (*Входные данные:     *)
 Funct:Integer;  {Вариант задания (1..11)}
 (*Выходные данные:    *)
 X:MatrType;     {Матрица собственных векторов}
 L:VectorType;   {Вектор собственных чисел}
 (*Вспомогательные переменные*)
 N:Integer;      {Размерность матрицы}
 i,j,Error:Integer;
 myFile:Text;
Begin
 if ParamCount<>0 then
  begin
   {Считываем параметры, заданные с помощью командной стро-
ки}
   Val(ParamStr(1),Funct,Error);
   {Создаем файл результатов}
   Assign(myFile,'results.txt');
   ReWrite(myFile);
  end
 else
  begin
   {Считываем параметры с клавиатуры}
   WriteLn;
   Write('Введите вариант задания (1..11): '); ReadLn(Funct);
   AssignCRT(myFile);
   ReWrite(myFile);
```

```
  end;
  if Funct=1 then
   N:=4
  else
   N:=3;
  Lab3(TestMatr[Funct],N,L,X); {вычисление собственных чисел и
векторов}
  {Вывод результатов}
  WriteLn(myFile,'Функция:             ',Funct);
  WriteLn(myFile,'Собственные числа:');
  for i:=1 to N do
   WriteLn(myFile,L[i]);
  WriteLn(myFile,'Собственные векторы:');
  for i:=1 to N do
   begin
    for j:=1 to N do
     Write(myFile,X[i,j],' ');
    WriteLn(myFile);
   end;
End.
```

ЛАБОРАТОРНАЯ РАБОТА № 5

```
{Интерполирование и численное дифференцирование функций:
шаблон программы}
{Программа написана для компиляторов Turbo Pascal/Borland
Pascal фирмы Borland inc.}
uses Crt;

type
 IntervalType=array[1..2,0..1000] of Real;  {Сетка интерполяции;}
     {1-я строка - значения узлов в сетке интерполяции}
     {2-я строка - значения функции, вычисленной в узлах интерпо-
ляции}
 IntervalOutputType=array[1..4,0..1000] of Real;
     {1-я строка - значения узлов в сетке интерполяции}
     {2-я строка - значения функции, вычисленной в узлах интерпо-
ляции}
     {3-я строка - значения первой производной функции}
```

```
    {4-я строка - значения второй производной функции}
Function Pow(A,B: Real):Real; {Вычисляет степень A^B}
 function relerror(observed, actual: real):real;
  begin
   if actual=0.0 then relerror:=abs(observed)
   else relerror:=abs(observed/actual-1);
  end;
var
 i:Byte;
 Res:Real;
begin
  if (Trunc(b)=b) and (b>0) then
   begin
    Res:=1;
    for i:=1 to Trunc(b) do
     Res:=Res*a;
    pow:=Res;
    Exit;
   end;
  if a=0 then
    if b=0 then pow:=1                    { 0^0 = 1 }
    else if b<0 then
    else pow:=0                           { 0^x = 0 }
  else if a<0 then
    if abs(b)<1e-10 then pow:=1
    else if relerror(b,round(b))<1e-8 then
      pow:=(1-2*ord(odd(round(b))))*exp(b*ln(abs(a)))
    else if (relerror(1/b,round(1/b))<1e-8) and odd(round(1/b)) then
      pow:=-exp(b*ln(abs(a)))
    else
  else pow:=exp(b*ln(a))
end;

Function F(N:Integer;X:Real):Real; (*Вычисляет значение функции
                                    из варианта задания
                                    с номером N в точке X          *)
  Begin
   case N of
```

```
    1: F:=Exp(-X/2);
    2: F:=Pow(X,1/3);
    3: F:=Exp(-Sqr(X-5));
    4: F:=Exp(-Sqr(X-3))+Exp(-Sqr(X-5));
    5: F:=Exp(Sqr(X));
    6: F:=X*Ln(X);
    7: F:=1/(1-Pow(X,3));
    8: F:=Ln(X-1);
    9: F:=Sqrt(X)+1;
    10: F:=Sqr(Sin(X))+1;
    11: F:=1-Sqr(Cos(X));
    12: F:=1/(1+Ln(X));
    13: F:=1/Sin(X);
    14: F:=2*Sin(X)/Cos(X);
    15: F:=3*Cos(X)/Sin(X);
    16: F:=Sin(X);
    17: F:=Cos(X);
   end;
  End;
Procedure Newton(N:Integer; A1,B1:Real; N1:Integer;
            var X:IntervalType; var Y:IntervalOutputType);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для интерполяции
   полиномом Ньютона
   Входные параметры:
    N:Integer      - количество узлов в исходной сетке;
    A1,B1:Real     - границы сетки интерполирования;
    N1:Integer     - количество узлов сетки интерполировния ([0..N1]);
    X:IntervalType  - исходная сетка узлов
                 ( X[1,i] - значения узлов;
                   X[2,i] - значения функции в этих узлах )
   Выходные параметры, возвращаемые процедурой:
    Y:IntervalType  - сетка интерполяции
                 ( Y[1,i] - значения узлов сетки интерполяции;
                   Y[2,i] - значения функции в этих узлах;
                   Y[3,i] - значения первой производной;
```

```
                 Y[4,i] - значения второй производной )   *)
 End;

Procedure Lagrange(N:Integer; A1,B1:Real; N1:Integer;
           var X:IntervalType; var Y:IntervalOutputType);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для ин-
терполяции
   полиномом Лагранжа
   Входные параметры:
    N:Integer       -  количество узлов в исходной сетке;
    A1,B1:Real      -  границы сетки интерполирования;
    N1:Integer           -   количество узлов сетки интерполировния
([0..N1]);
    X:IntervalType   -  исходная сетка узлов
                ( X[1,i] - значения узлов;
                  X[2,i] - значения функции в этих узлах )
   Выходные параметры, возвращаемые процедурой:
    Y:IntervalType   -  сетка интерполяции
                ( Y[1,i] - значения узлов сетки интерполяции;
                  Y[2,i] - значения функции в этих узлах;
                  Y[3,i] - значения первой производной;
                  Y[4,i] - значения второй производной )   *)
 End;

Procedure Lab4(Funct:Integer; A,B:Real; N:Integer; A1,B1:Real;
N1:Integer;
          Method:Integer; var Y:IntervalOutputType);
 var
  i:Integer;
  h:Real;
  X:^IntervalType;
 Begin
  New(X);
  h:=(B-A)/N;
  {Заполняем исходную сетку:}
  for i:=0 to N do
   begin
```

```
    X^[1,i]:=A+h*i;
    X^[2,i]:=F(Funct,X^[1,i]);
   end;
  case Method of
   1: Newton(N,A1,B1,N1,X^,Y);
   2: Lagrange(N,A1,B1,N1,X^,Y);
  end;
  Dispose(X);
 End;

var
 (*Входные данные:      *)
 Funct:Integer;  {Вариант задания (1..17)}
 A,B:Real;       {Границы сетки задания функции}
 N:Integer;      {Количество узлов сетки задания функции}
 A1,B1:Real;     {Границы сетки интерполирования}
 N1:Integer;     {Количество узлов сетки интерполирования}
 Method:Integer; {Метод приближения}
 (*Выходные данные:     *)
 Y:IntervalOutputType; {Сетка интерполирования}
 (*Вспомогательные переменные*)
 i,Error:Integer;
 myFile:Text;
Begin
 if ParamCount<>0 then
  begin
   {Считываем параметры, заданные с помощью командной стро-
ки}
   Val(ParamStr(1),Funct,Error);
   Val(ParamStr(2),A,Error);
   Val(ParamStr(3),B,Error);
   Val(ParamStr(4),N,Error);
   Val(ParamStr(5),A1,Error);
   Val(ParamStr(6),B1,Error);
   Val(ParamStr(7),N1,Error);
   Val(ParamStr(9),Method,Error);
   {Создаем файл результатов}
   Assign(myFile,'results.txt');
```

```
   ReWrite(myFile);
  end
 else
  begin
   {Считываем параметры с клавиатуры}
   WriteLn;
   Write('Введите вариант задания (1..17): '); ReadLn(Funct);
   Write('Введите нижнюю границу интервала задания функции: ');
ReadLn(A);
   Write('Введите верхнюю границу интервала задания функции: ');
ReadLn(B);
   Write('Введите количество узлов интервала задания функции: ');
ReadLn(N);
   Write('Введите нижнюю границу интервала интерполяции: ');
ReadLn(A1);
   Write('Введите верхнюю границу интервала интерполяции: ');
ReadLn(B1);
   Write('Введите количество узлов интервала интерполяции: ');
ReadLn(N1);
   Write('Введите номер метода вычислений (1 - полином Ньютона;
',
     '2 - полином Лагранжа):');  ReadLn(Method);
   AssignCRT(myFile);
   ReWrite(myFile);
  end;
  Lab4(Funct,A,B,N,A1,B1,N1,Method,Y); {Интерполяция}
  {Вывод результатов}
  WriteLn(myFile,'Функция:            ',Funct);
  WriteLn(myFile,'Метод:              ',Method);
  for i:=0 to N1 do
   WriteLn(myFile,'X',i,':  ',Y[1,i],'    Y',i,':  ',Y[2,i],'    Y',i,#39,':  ',Y[3,i],'
Y',i,#39#39,': ',Y[4,i]);
End.
```

ЛАБОРАТОРНАЯ РАБОТА № 6

{Приближение сплайнами: шаблон программы}
{Программа написана для компиляторов Turbo Pascal/Borland Pascal фирмы Borland inc.}

```
uses Crt;

type
 IntervalType=array[1..2,0..1000] of Real;  {Сетка интерполяции;}
     {1-я строка - значения узлов в сетке интерполяции}
     {2-я строка - значения функции, вычисленной в узлах интерпо-
ляции}

Function Pow(A,B: Real):Real; {Вычисляет степень A^B}
 function relerror(observed, actual: real):real;
  begin
   if actual=0.0 then relerror:=abs(observed)
   else relerror:=abs(observed/actual-1);
  end;
var
 i:Byte;
 Res:Real;
begin
  if (Trunc(b)=b) and (b>0) then
   begin
    Res:=1;
    for i:=1 to Trunc(b) do
     Res:=Res*a;
    pow:=Res;
    Exit;
   end;
  if a=0 then
    if b=0 then pow:=1                            { 0^0 = 1 }
    else if b<0 then
    else pow:=0                              { 0^x = 0 }
  else if a<0 then
    if abs(b)<1e-10 then pow:=1
    else if relerror(b,round(b))<1e-8 then
      pow:=(1-2*ord(odd(round(b))))*exp(b*ln(abs(a)))
    else if (relerror(1/b,round(1/b))<1e-8) and odd(round(1/b)) then
      pow:=-exp(b*ln(abs(a)))
    else
  else pow:=exp(b*ln(a))
```

```
end;

Function F(N:Integer;X:Real):Real; (*Вычисляет значение функции
                                   из варианта задания
                                   с номером N в точке X              *)
  Begin
   case N of
    1: F:=Sin(X);
    2: F:=Cos(X);
    3: F:=Sqr(X);
    4: F:=Exp(-Sqr(X));
    5: F:=Pow(X,4)-4*Pow(X,3)-40*Sqr(X)-56*X-20;
    6: F:=X+Ln(X);
    7: F:=Pow(X,4)-4*Pow(X,3)-40*Sqr(X)-56*X-20;
    8: F:=Pow(X,3)-Sqr(X)-5*X-3;
    9: F:=Pow(X,3)-Sqr(X)-4*X-4;
    10,11: F:=Exp(X)-1/X;
   end;
  End;

Procedure Linear(N:Integer; A1,B1:Real; N1:Integer;
               var X,Y:IntervalType);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для при-
ближения
   линейными сплайнами
   Входные параметры:
    N:Integer        -  количество узлов в исходной сетке;
    A1,B1:Real       -  границы сетки интерполирования;
    N1:Integer            -   количество  узлов  сетки  интерполировния
([0..N1]);
    X:IntervalType   -  исходная сетка узлов
                      ( X[1,i] - значения узлов;
                        X[2,i] - значения функции в этих узлах )
   Выходные параметры, возвращаемые процедурой:
    Y:IntervalType   -  сетка интерполяции
                      ( Y[1,i] - значения узлов сетки интерполяции;
                        Y[2,i] - значения функции в этих узлах )   *)
```

```
 End;

Procedure Parabol(N:Integer; A1,B1:Real; N1:Integer; Tp:Integer;
            var X,Y:IntervalType);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для при-
ближения
   параболическими сплайнами
   Входные параметры:
    N:Integer       -  количество узлов в исходной сетке;
    A1,B1:Real      -  границы сетки интерполирования;
    N1:Integer         -   количество  узлов  сетки  интерполировния
([0..N1]);
    Tp:Integer      -  тип граничных условий (1 - левый; 2 - правый);
    X:IntervalType   -  исходная сетка узлов
                   ( X[1,i] - значения узлов;
                     X[2,i] - значения функции в этих узлах )
   Выходные параметры, возвращаемые процедурой:
    Y:IntervalType   -  сетка интерполяции
                   ( Y[1,i] - значения узлов сетки интерполяции;
                     Y[2,i] - значения функции в этих узлах )   *)
 End;

Procedure Cube(N:Integer; A1,B1:Real; N1:Integer; Tp:Integer;
            var X,Y:IntervalType);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для при-
ближения
   кубическими сплайнами
   Входные параметры:
    N:Integer       -  количество узлов в исходной сетке;
    A1,B1:Real      -  границы сетки интерполирования;
    N1:Integer         -   количество  узлов  сетки  интерполировния
([0..N1]);
    Tp:Integer      -  тип граничных условий (1 - левый; 2 - правый);
    X:IntervalType   -  исходная сетка узлов
                   ( X[1,i] - значения узлов;
                     X[2,i] - значения функции в этих узлах )
```

```
  Выходные параметры, возвращаемые процедурой:
   Y:IntervalType   -  сетка интерполяции
                    ( Y[1,i] - значения узлов сетки интерполяции;
                      Y[2,i] - значения функции в этих узлах )   *)
 End;

Procedure Lab5(Funct:Integer; A,B:Real; N:Integer; A1,B1:Real;
N1:Integer;
           Tp,Method:Integer; var Y:IntervalType);
 var
  i:Integer;
  h:Real;
  X:^IntervalType;
 Begin
  New(X);
  h:=(B-A)/N;
  {Заполняем исходную сетку:}
  for i:=0 to N do
   begin
    X^[1,i]:=A+h*i;
    X^[2,i]:=F(Funct,X^[1,i]);
   end;
  case Method of
   1: Linear(N,A1,B1,N1,X^,Y);
   2: Parabol(N,A1,B1,N1,Tp,X^,Y);
   3: Cube(N,A1,B1,N1,Tp,X^,Y);
  end;
  Dispose(X);
 End;

var
 (*Входные данные:      *)
 Funct:Integer;  {Вариант задания (1..11)}
 A,B:Real;       {Границы сетки задания функции}
 N:Integer;      {Количество узлов сетки задания функции}
 A1,B1:Real;     {Границы сетки интерполирования}
 N1:Integer;     {Количество узлов сетки интерполирования}
```

```
 Tp:Integer;      {Тип граничных условий (для параболических и ку-
бических сплайнов)}
 Method:Integer; {Метод приближения}
 (*Выходные данные:     *)
 Y:IntervalType; {Сетка интерполирования}
 (*Вспомогательные переменные*)
 i,Error:Integer;
 myFile:Text;
Begin
 if ParamCount<>0 then
  begin
   {Считываем параметры, заданные с помощью командной стро-
ки}
   Val(ParamStr(1),Funct,Error);
   Val(ParamStr(2),A,Error);
   Val(ParamStr(3),B,Error);
   Val(ParamStr(4),N,Error);
   Val(ParamStr(5),A1,Error);
   Val(ParamStr(6),B1,Error);
   Val(ParamStr(7),N1,Error);
   Val(ParamStr(8),Tp,Error);
   Val(ParamStr(9),Method,Error);
   {Создаем файл результатов}
   Assign(myFile,'results.txt');
   ReWrite(myFile);
  end
 else
  begin
   {Считываем параметры с клавиатуры}
   WriteLn;
   Write('Введите вариант задания (1..11): '); ReadLn(Funct);
   Write('Введите нижнюю границу интервала задания функции: ');
ReadLn(A);
   Write('Введите верхнюю границу интервала задания функции: ');
ReadLn(B);
   Write('Введите количество узлов интервала задания функции: ');
ReadLn(N);
```

```
  Write('Введите нижнюю границу интервала интерполяции: ');
ReadLn(A1);
  Write('Введите верхнюю границу интервала интерполяции: ');
ReadLn(B1);
  Write('Введите количество узлов интервала интерполяции: ');
ReadLn(N1);
  Write('Введите тип граничных условий (1 - левый, 2 - правый): ');
ReadLn(Tp);
  Write('Введите номер метода вычислений (1 - линейные сплайны;
',
   '2 - параболические; 3 - кубические сплайны):');
ReadLn(Method);
  AssignCRT(myFile);
  ReWrite(myFile);
 end;
 Lab5(Funct,A,B,N,A1,B1,N1,Tp,Method,Y); {Приближение сплай-
нами}
 {Вывод результатов}
 WriteLn(myFile,'Функция:          ',Funct);
 WriteLn(myFile,'Метод:            ',Method);
 for i:=0 to N1 do
  WriteLn(myFile,'X',i,': ',Y[1,i],'     Y',i,': ',Y[2,i]);
End.
```

ЛАБОРАТОРНАЯ РАБОТА № 7

```
{Численное интегрирование функций: шаблон программы}
{Программа написана для компиляторов Turbo Pascal/Borland
Pascal фирмы Borland inc.}
uses Crt;

Function Pow(A,B: Real):Real; {Вычисляет степень A^B}
 function relerror(observed, actual: real):real;
  begin
   if actual=0.0 then relerror:=abs(observed)
   else relerror:=abs(observed/actual-1);
  end;
var
 i:Byte;
```

```
 Res:Real;
begin
  if (Trunc(b)=b) and (b>0) then
   begin
    Res:=1;
    for i:=1 to Trunc(b) do
     Res:=Res*a;
    pow:=Res;
    Exit;
   end;
  if a=0 then
    if b=0 then pow:=1                          { 0^0 = 1 }
    else if b<0 then
    else pow:=0                             { 0^x = 0 }
  else if a<0 then
    if abs(b)<1e-10 then pow:=1
    else if relerror(b,round(b))<1e-8 then
      pow:=(1-2*ord(odd(round(b))))*exp(b*ln(abs(a)))
    else if (relerror(1/b,round(1/b))<1e-8) and odd(round(1/b)) then
      pow:=-exp(b*ln(abs(a)))
    else
  else pow:=exp(b*ln(a))
end;

Function F(N:Integer;X:Real):Real; (*Вычисляет значение подинте-
гральной
                                    функции из вариантов заданий
                                    с номером N в точке X              *)
  Begin
   case N of
    1: F:=1/Sqrt(5+4*X-Sqr(X));
    2: F:=X*X*X/(Sqr(Sqr(X))*Sqr(X)+1);
    3: F:=X/(Sqr(X)+3*X+2);
    4: F:=1/(Sqr(X)+4);
    5: F:=(1+Sqrt(X))/Sqr(X);
    6: F:=Sqrt(1+X*X*X);
    7: F:=Sqrt(1+Pow(X,5));
    8: F:=1/Sqrt(1+Pow(X,4));
```

```
   9: F:=1/Sqrt(1-Pow(X,4));
   10: F:=1/Pow(1+Sqr(X),1/3);
   11: F:=Sqrt(X*(1-X));
   12: F:=X*Ln(1+X);
   13: F:=Exp(Sqr(X));
   14: F:=Exp(X*X*X);
   15: F:=Exp(Sqrt(X));
   16: F:=1/Ln(X);
   17: F:=Ln(1+Sqr(X))/(1+Sqr(X));
   18: F:=Ln(5+4*Cos(X));
  end;
 End;

Procedure Rectangle(Funct:Integer; A,B,E:Real; var Y:Real; var N1:Integer);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для численного
    интегрирования функций методом прямоугольников.
   Входные параметры:
    Funct:Integer   -  вариант задания;
    A,B:Real        -  интервал интегрирования;
    E:Real          -  заданная точность.
   Выходные параметры, возвращаемые процедурой:
    Y:Real          -  значение интеграла;
    N1:Integer      -  число узлов;                    *)
 End;

Procedure Trapecy(Funct:Integer; A,B,E:Real; var Y:Real; var N1:Integer);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для численного
    интегрирования функций методом трапеций.
   Входные параметры:
    Funct:Integer   -  вариант задания;
    A,B:Real        -  интервал для поиска корней;
    E:Real          -  заданная точность.
```

  Выходные параметры, возвращаемые процедурой:
   Y:Real   - значение интеграла;
   N1:Integer   - число узлов;   *)
 End;

Procedure Simpson(Funct:Integer; A,B,E:Real; var Y:Real; var N1:Integer);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для численного
   интегрирования функций методом Симпсона.
  Входные параметры:
   Funct:Integer   - вариант задания;
   A,B:Real   - интервал интегрирования;
   E:Real   - заданная точность.
  Выходные параметры, возвращаемые процедурой:
   Y:Real   - значение интеграла;
   N1:Integer   - число узлов;   *)
 End;

Procedure Splayn(Funct:Integer; A,B,E:Real; var Y:Real; var N1:Integer);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для численного
   интегрирования функций по квадратурной формуле на основе
   кубических сплайнов.
  Входные параметры:
   Funct:Integer   - вариант задания;
   A,B:Real   - интервал интегрирования;
   E:Real   - заданная точность;
  Выходные параметры, возвращаемые процедурой:
   Y:Real   - значение интеграла;
   N1:Integer   - число узлов;   *)
 End;

Procedure Chebyshev(Funct:Integer; A,B,E:Real; P:Integer; var Y:Real; var N1:Integer);

```
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для чис-
ленного
    интегрирования функций по квадратурной формуле Чебышева.
   Входные параметры:
    Funct:Integer    -  вариант задания;
    A,B:Real         -  интервал интегрирования;
    E:Real           -  заданная точность;
    P:Integer        -  порядок формулы.
   Выходные параметры, возвращаемые процедурой:
    Y:Real           -  значение интеграла;
    N1:Integer       -  число узлов;                    *)
 End;

Procedure Gauss(Funct:Integer; A,B,E:Real; P:Integer; var Y:Real; var
N1:Integer);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для чис-
ленного
    интегрирования функций по квадратурной формуле Гаусса.
   Входные параметры:
    Funct:Integer    -  вариант задания;
    A,B:Real         -  интервал интегрирования;
    E:Real           -  заданная точность;
    P:Integer        -  порядок формулы.
   Выходные параметры, возвращаемые процедурой:
    Y:Real           -  значение интеграла;
    N1:Integer       -  число узлов;                    *)
 End;

Procedure Lab6(Funct:Integer; A,B:Real; E:Real; P:Integer; Meth-
od:Integer;
            var Y:Real; var N1:Integer);
 Begin
  case Method of
   1:Rectangle(Funct,A,B,E,Y,N1);
   2:Trapecy(Funct,A,B,E,Y,N1);
   3:Simpson(Funct,A,B,E,Y,N1);
```

```
   3:Splayn(Funct,A,B,E,Y,N1);
   5:Chebyshev(Funct,A,B,E,P,Y,N1);
   6:Gauss(Funct,A,B,E,P,Y,N1);
  end;
 End;


var
 (*Входные данные:      *)
 Funct:Integer;  {Вариант задания}
 A,B:Real;       {Границы интервала}
 N:Integer;      {Количество узлов}
 E:Real;         {Относительная точность}
 P:Integer;      {Порядок формулы (для формул Гаусса и Чебышева)}
 Method:Integer; {Метод интегрирования}
 (*Выходные данные:     *)
 Y:Real;         {Значение интеграла}
 N1:Integer;     {Количество узлов}
 (*Вспомогательные переменные*)
 i,Error:Integer;
 myFile:Text;
Begin
 if ParamCount<>0 then
  begin
   {Считываем параметры, заданные с помощью командной стро-
ки}
   Val(ParamStr(1),Funct,Error);
   Val(ParamStr(2),A,Error);
   Val(ParamStr(3),B,Error);
   Val(ParamStr(4),N,Error);
   Val(ParamStr(5),E,Error);
   Val(ParamStr(6),P,Error);
   Val(ParamStr(7),Method,Error);
   {Создаем файл результатов}
   Assign(myFile,'results.txt');
   ReWrite(myFile);
  end
 else
```

```
  begin
   {Считываем параметры с клавиатуры}
   WriteLn;
   Write('Введите вариант задания (1..18): '); ReadLn(Funct);
   Write('Введите нижнюю границу интервала: '); ReadLn(A);
   Write('Введите верхнюю границу интервала: '); ReadLn(B);
   Write('Введите количество узлов: '); ReadLn(N);
   Write('Введите относительную точность: '); ReadLn(E);
   Write('Введите порядок формулы (для формул Гаусса и Чебыше-
ва): '); ReadLn(P);
   Write('Введите номер метода вычислений (1 - прямоугольников;
2 - трапеций;',
     '3 - Симпсона; 4 - сплайнов; 5 - Чебышева; 6 - Гаусса) :');
   ReadLn(Method);
   AssignCRT(myFile);
   ReWrite(myFile);
  end;
  Lab6(Funct,A,B,E,P,Method,Y,N1); {Численное интегрирование}
  {Вывод результатов}
  WriteLn(myFile,'Функция:            ',Funct);
  WriteLn(myFile,'Метод:              ',Method);
  WriteLn(myFile,'Значение интеграла:   ',Y);
  WriteLn(myFile,'Количество узлов:     ',N1);
End.
```

ЛАБОРАТОРНАЯ РАБОТА № 8

```
{Дифференциальные уравнения первого порядка: шаблон про-
граммы}
{Программа написана для компиляторов Turbo Pascal/Borland
Pascal фирмы Borland inc.}
uses Crt;

type
 ResType=array[1..2,1..1000] of Real;

Function F(N:Integer;X,Y:Real):Real; (*Вычисляет значение первой
производной
                                      Y в точках X,Y для варианта задания N.
```

```
                              Пользуйтесь этой функцией для
                              вычислений                    *)
 Begin
  case N of
   1: F:=X*Y*Y*Y-Y;
   2: F:=Exp(X)*Sqr(Y)-2*Y;
   3: F:=Sqr(X)+Sqr(Y);
   4: F:=Exp(X)-X;
   5: F:=X*Exp(-Sqr(X));
   6: F:=Exp(2*X)-Y*cos(X);
   7: F:=Sin(2*X)-Y*Sin(X)/Cos(X);
   8: F:=Sqr(Y)+Y*sin(2*X)+cos(2*X);
   9: F:=X*Sqr(Y)+Y;
   10: F:=Exp(X-Y)-Exp(X);
  end;
 End;

Procedure DigIntegral(Funct:Integer; Y0,A,B,h,E:Real;
               var Y:ResType; var N:Integer);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для
   решения дифференциальных уравнений первого порядка мето-
дом
   численного интегрирования.
  Входные параметры:
   Funct:Integer    -  вариант задания;
   Y0               -  значения функции Y(0);
   A,B:Real         -  интервал дифференцировния;
   h:Real           -  шаг интервала дифференцирования;
   E:Real           -  заданная точность.
  Выходные параметры, возвращаемые процедурой:
   Y:ResType        -  двумерный массив результатов дифференциро-
вания, где
                Y[1,i] - значения узлов сетки;
                Y[2,i] - значения функции Y в этих узлах;
   N:Integer        -  число узлов в массиве Y                  *)
 End;
```

```
Procedure RungeCutt(Funct:Integer; Y0,A,B,h,E:Real; P:Integer;
                 var Y:ResType; var N:Integer);
 Begin
  (*В теле этой процедуры напишите свою подпрограмму для
    решения дифференциальных уравнений первого порядка по
    схеме Рунге-Кутта первого или четвертого порядка.
   Входные параметры:
    Funct:Integer    -  вариант задания;
    Y0               -  значения функции Y(0);
    A,B:Real         -  интервал дифференцировния;
    h:Real           -  шаг интервала дифференцирования;
    E:Real           -  заданная точность;
    P:Integer        -  порядок схемы (1 или 4).
   Выходные параметры, возвращаемые процедурой:
    Y:ResType        -  двумерный массив результатов дифференциро-
вания, где
                     Y[1,i] - значения узлов сетки;
                     Y[2,i] - значения функции Y в этих узлах;
    N:Integer        -  число узлов в массиве Y                    *)
 End;


Procedure Lab7(Funct:Integer; Y0,A,B,h,E:Real; P,Method:Integer;
             var Y:ResType; var N:Integer);
 Begin
  case Method of
   1:DigIntegral(Funct,Y0,A,B,h,E,Y,N);
   2:RungeCutt(Funct,Y0,A,B,h,E,N,Y,N);
  end;
 End;


var
 (*Входные данные:      *)
 Funct:Integer;  {Вариант задания}
 Y0:Real;        {Значение функции Y(0)}
 A,B:Real;       {Границы интервала}
 h:Real;         {Количество узлов}
```

```
 E:Real;        {Относительная точность}
 P:Integer;     {Порядок формулы (для метода Рунге-Кутта)}
 Method:Integer; {Метод решения уравнения}
 (*Выходные данные:    *)
 Y:ResType;     {Значение функции Y на сетке}
 N:Integer;     {Количество узлов сетки}
 (*Вспомогательные переменные*)
 i,Error:Integer;
 myFile:Text;
Begin
 if ParamCount<>0 then
  begin
   {Считываем параметры, заданные с помощью командной стро-
ки}
   Val(ParamStr(1),Funct,Error);
   Val(ParamStr(2),Y0,Error);
   Val(ParamStr(3),A,Error);
   Val(ParamStr(4),B,Error);
   Val(ParamStr(5),h,Error);
   Val(ParamStr(6),E,Error);
   Val(ParamStr(7),P,Error);
   Val(ParamStr(8),Method,Error);
   {Создаем файл результатов}
   Assign(myFile,'results.txt');
   ReWrite(myFile);
  end
 else
  begin
   {Считываем параметры с клавиатуры}
   WriteLn;
   Write('Введите вариант задания (1..10): '); ReadLn(Funct);
   Write('Введите значение граничного условия Y(0): '); ReadLn(Y0);
   Write('Введите нижнюю границу интервала: '); ReadLn(A);
   Write('Введите верхнюю границу интервала: '); ReadLn(B);
   Write('Введите шаг интервала: '); ReadLn(h);
   Write('Введите относительную точность: '); ReadLn(E);
   Write('Введите порядок формулы (для метода Рунге-Кутта, 1 или
4): '); ReadLn(P);
```

```
  Write('Введите номер метода вычислений (1 - численное интегрирование; ',
      '2 - Рунге-Кутта) :');ReadLn(Method);
  AssignCRT(myFile);
  ReWrite(myFile);
 end;
 Lab7(Funct,Y0,A,B,h,E,P,Method,Y,N); {Решение диф. уравнений}
 {Вывод результатов}
 WriteLn(myFile,'Функция:           ',Funct);
 WriteLn(myFile,'Метод:             ',Method);
 for i:=1 to N do
  WriteLn(myFile,'Y (',Y[1,i],') = ',Y[2,i]);
End.
```