

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Уфимский государственный нефтяной технический
университет»**

Кафедра прикладных и естественнонаучных дисциплин

**ТИПОВЫЕ ЗАДАНИЯ
К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ
И ПРАКТИЧЕСКИХ РАБОТ ПО ИНФОРМАТИКЕ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

**Уфа
Издательство УГНТУ
2017**

В учебно-методическом пособии представлен комплекс лабораторных и практических работ по дисциплине "Информатика".

В пособии приведены задания для десяти работ, посвященных изучению основ алгоритмизации, программирования на языке Паскаль и решения функциональных задач. Для каждого задания кратко изложен необходимый теоретический материал, указан порядок выполнения работы и требования к ее оформлению, показаны примеры решения, приведены контрольные вопросы для подготовки студентов к защите отчета по выполненной работе.

Пособие разработано для студентов, обучающихся по направлениям (специальностям): 08.03.01 "Строительство", 05.03.01 "Геология", 21.05.02 "Прикладная геология", 21.05.09 "Технология геологической разведки".

Составители: Виноградов Д.А., доц., канд. техн. наук
 Ращепкин А. К., доц., канд. техн. наук
 Салагаев В.Б., доц., канд. техн. наук

Рецензент Юнусов Р.З., доц., канд. техн. наук

ВВЕДЕНИЕ

Пособие содержит задания к лабораторным (практическим) работам по курсу информатики базовой части образовательной программы. Задания 1 – 5 посвящены изучению основ алгоритмизации и программирования. В качестве учебной среды программирования выбрана система PascalABC.NET, предназначенная для обучения программированию на алгоритмическом языке Паскаль. Работы направлены на изучение принципов построения базовых типов алгоритмов, формирования у студентов практических навыков программирования, готовности использовать разнообразные приемы, методы и средства, необходимые в процессе разработки программного обеспечения.

При выполнении каждой работы нужно разработать алгоритм и программу решения задачи в соответствии с вариантом задания, указанным преподавателем. В пособии приведены краткие сведения о конструкциях языка программирования Паскаль, используемых в данных работах, показаны примеры выполнения заданий и оформления отчетов по ним.

Порядок выполнения первых пяти работ следующий:

1. Проанализировать и понять формулировку задания. Определить все переменные задачи и задать типы данных. Выяснить, какие из переменных являются исходными данными задачи, а какие - результатами ее решения, определить последовательность решения задачи, подготовить тестовый расчет (контрольный пример).
2. Разработать алгоритм решения поставленной задачи в виде последовательности действий исполнителя (компьютера) по обработке данных. Алгоритм представить в виде графической блок-схемы. Алгоритм должен приводить к желаемому результату при любых допустимых для данной задачи значениях исходных данных (предусмотреть ввод исходных данных с клавиатуры).
3. Представить разработанный алгоритм в виде программы на алгоритмическом языке Паскаль. В программе предусмотреть ввод исходных данных в форме диалога «пользователь – программа» и вывод данных в наглядной форме.
4. Ввести программу в среде разработки PascalABC, выполнить отладку программы, исправить все логические и синтаксические ошибки (если таковые имеются), подтвердить правильность программы тестовыми расчётами.
5. Выполнить рабочий расчет по программе, используя исходные данные из своего варианта задания.
6. Подготовить отчет по выполненному заданию. Продемонстрировать преподавателю работу программы на компьютере, оформить отчет и защитить его.

Работы 6 – 10 посвящены численному решению на компьютере прикладных вычислительных задач, часто встречающихся в инженерной практике. Представлены задачи вычисления корней алгебраических и трансцендентных уравнений, решения систем линейных и нелинейных уравнений, вычисления простых и кратных интегралов. Работы направлены на изучение соответствующих численных вычислительных методов, алгоритмов для их реализации на компьютере, получения навыков применения программ инженерного анализа.

Каждое задание включает постановку задачи, краткую теоретическую часть, индивидуальные варианты заданий, порядок его выполнения. Задание поясняется разобранными примерами, приводятся описания алгоритмов и программы на языке Паскаль, необходимые для решения поставленной задачи. В конце каждого задания указаны контрольные вопросы для самоконтроля студентами своих знаний, приведены требования к содержанию и оформлению отчетов. Аккуратно оформленный отчет по каждой работе студент предоставляет преподавателю для его защиты.

СРЕДА ПРОГРАММИРОВАНИЯ «PascalABC.NET»

Среда программирования или интегрированная среда разработки (англ. IDE, Integrated development environment) - система программных средств, используемая для разработки программного обеспечения. Среда разработки содержит в себе все необходимое для разработки программ:

- текстовый редактор с подсветкой синтаксиса конкретного языка программирования (в нем набирается исходный программный код);
- компилятор или интерпретатор - транслирует программу, написанную на высокоуровневом языке программирования в машинный язык (машинный код), непосредственно понятный компьютеру;
- отладчик служит для тестирования программы, выявления в ней логических ошибок и отладки программ;
- средства автоматизации сборки готовой программы (исполняемый программный код);
- справочную систему.

Система PascalABC.NET разработана на факультете математики, механики и компьютерных наук Южного федерального университета. PascalABC.NET - это бесплатная, простая и мощная среда разработки для языка программирования Паскаль, ориентированная на обучение программированию, опирающаяся на передовую платформу программирования Microsoft.NET.

Официальный сайт в сети Интернет, с которого можно скачать дистрибутив этой среды, расположен по адресу <http://pascalabc.net/>. Система имеет удобный графический интерфейс, имеется функция подсветки синтаксиса, поддерживает многооконный режим работы, результаты работы отображаются в нижней части окна. Предусмотрены возможности генерации исполняемых файлов, запуск программы вне программной оболочки, пошагового выполнения программы, просмотра и редактирования текущих значений переменных и другие современные возможности. Для запуска программы на исполнение можно воспользоваться кнопкой «Выполнить» на панели инструментов, или нажать клавишу F9.

Подробное описание системы PascalABC.NET, большое количество примеров программ можно найти на сайте: <http://pascalabc.net/primeri-programm>.

1. РАБОТА №1. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЛИНЕЙНОЙ СТРУКТУРЫ

1.1. ЦЕЛЬ РАБОТЫ

Формирование у студентов умения анализировать исходные данные задачи, навыков разработки и программирования вычислительного процесса линейной структуры. Изучение системы программирования PascalABC.NET, организации работы по набору, отладке, компиляции, запуска программы на исполнение.

1.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

Логическая структура любого алгоритма может быть представлена комбинацией трех базовых структур: линейной, разветвленной, циклической.

Линейным называется алгоритм, в котором результат получается путем однократного выполнения заданной последовательности действий при любых значениях исходных данных. Операторы линейной программы выполняются последовательно, один за другим, в соответствии с их расположением в программе.

Программа на языке Паскаль имеет следующую структуру:

```
Program <Имя программы>;
<Раздел описаний>
Begin
<Тело программы>
End.
```

Здесь слова *Program*, *Begin* и *End* являются служебными. Все объекты, не являющиеся зарезервированными в Паскале, перед первым использованием в программе должны быть описаны (объявлены в разделе описаний). В разделе описаний, если нужно, создаются подразделы для описания разных объектов программы, например, подраздел описания переменных может выглядеть так:

```
Var b, m, alfa : real; c5, day, year : integer;
```

Переменная в Паскале – это именованный участок памяти для хранения данных определенного типа. Тип данных задается служебными словами: *real* – вещественный, *integer* – целый, *string* – строковый, *boolean* – логический, и др.

Тело программы представляет собой последовательность операторов (команд), реализующих алгоритм решения поставленной задачи. В данной работе для написания программы требуются следующие команды:

- **read**(<Список ввода>), **readln**(<Список ввода>) – вызов стандартной процедуры ввода данных (*read* и *readln* при вводе данных с клавиатуры работают одинаково), например, команда **read(x, y)** вводит с клавиатуры два значения и записывает их в ячейки памяти для переменных *x* и *y*;
- **write**(<Список вывода>), **writeln**(<Список вывода>) – вызов стандартной процедуры вывода данных на экран монитора (после выполнения *writeln* происходит переход на новую строку, а после выполнения *write* – нет),

например, команда **write**(‘значение **x**=’, **x**) выводит на экран строку: **значение **x**=** и следом значение из ячейки памяти переменной **x**;

- <Имя переменной> := <Выражение> – присваивание значения переменной, например, оператор **x := a + b** вычисляет выражение **a + b** и полученный результат записывает в ячейку памяти **x**.

Выражения в Паскале представляют собой операнды (переменные, константы и функции), соединенные символами операций: арифметических (+, −, *, /, *div*, *mod*), логических (*not*, *and*, *or*) и операций сравнения (<, >, =, <>, <=, >=).

При составлении выражений часто используют стандартные математические функции из библиотеки системы PascalABC.NET: *sin*, *cos*, *tan*, *arcsin*, *arccos*, *arctan* – тригонометрические функции; *ln*, *log10* – натуральный и десятичный логарифмы; *sqr* – квадратный корень; *sqr* – возведение в квадрат; *exp* – экспонента числа; *abs* – модуль числа; *trunc* – преобразование дробного числа к целому; *power(x,y)* – возведение числа *x* в степень *y*.

Все команды языка Паскаль разделяются символом «;».

1.3. ЗАДАНИЯ К РАБОТЕ

Разработать алгоритм и программу вычисления значений заданных функций для произвольных значений исходных данных. После отладки программы выполнить контрольный тестовый расчет и сравнить результаты с ручным расчетом. Затем выполнить рабочий расчет для заданных по варианту значений исходных данных.

Варианты заданий

№ варианта	Заданные функции	Исходные данные
1	$z = (b + y^2) \cdot \frac{x+y/2}{y^2 \cdot (1+y^2)}$; $b = \sqrt[3]{\sin^2(\operatorname{tg}(y))}$; $x = \log_2(y)$	$y = 5,2$
2	$b = e^{ x-y } \cdot (\operatorname{tg}^2(z) + a)^x$; $a = \frac{\sqrt[3]{8 + x-y ^2} + 1}{x+y^2+2}$	$x = -3,8$; $y = 0,63$; $z = 0,91$
3	$a = \frac{y^{x+1}}{\sqrt[3]{ y-2,5 +3}} + \frac{z+\frac{y}{2 \cdot x}}{2 \cdot x+y }$; $z = (x+1)^{-\sin(y)}$	$x = 1,3$; $y = 21,7$
4	$a = \sqrt[3]{x + \sqrt[4]{b}}$; $b = \sqrt{ y } \cdot e^{-(y+t/2)}$; $t = \frac{\pi}{x \cdot y}$	$x = 25,4$; $y = -5,2$
5	$g = \left(\frac{x+3 \cdot a+y}{2 \cdot x} \right)^4 - \frac{\lg^2(x^4)}{x+3 \cdot a-y}$; $y = a^x$	$a = 6,45$; $x = 3,81$

6	$y = \sqrt[3]{x + \lg x + a }; x = (\sin^2(a^3) + \cos(a))^2; a = e^b + \frac{b}{2 \cdot \pi}$	$b = 0,75$
7	$y = \cos(b) \cdot \sqrt[3]{1 + \sin^2(b + a)}; b = \frac{a+c}{a \cdot c}; a = e^{-c} + c$	$c = 2,43$
8	$y = \frac{\sin(a \cdot b) + 1}{\sqrt{a + b \cdot \sin(a + b)}}; a = \ln\left(b + \lg^2\left(\frac{1}{b}\right)\right); b = 0,5 \cdot e^{1+c}$	$c = 6,3$
9	$c = \ln\left(x + \ln\left(x + \frac{ a }{3 \cdot x}\right)\right); x = 2,23 \cdot e^{2 \cdot d + 0,5}; d = \sin^2(b + a)$	$a = 0,5; b = 3$
10	$y = \cos(x) + \sqrt{1 + \sin^2(x + a)}; x = \ln(a + c); a = c + \frac{1}{2 \cdot c}$	$c = 3,2$
11	$y = \frac{1 + \sin(x)}{\cos(x) \sin(x)}; x = \sqrt[4]{1 + \ln^2(a + b)}; a = e^b + 1$	$b = 1,8$
12	$b = \cos(x) + \sqrt[3]{1 + \sin^2(x + c)}; x = \frac{a+c}{a \cdot c}; a = e^{-c} + c$	$c = 4,52$
13	$y = \frac{\sqrt{x^2 + a^2}}{2 \cdot \pi} \cdot \lg^2(x + a); x = e^{a+b}; a = (\sin(b) + \cos(b))^2$	$b = 5,4$
14	$y = \sin(x) \cdot \sqrt{x^2 + c^2}; x = \frac{c+a}{\lg(a)}; a = \sin^2\left(c + \frac{1}{2 \cdot c}\right)$	$c = 0,95$
15	$c = \left(\frac{x+3 \cdot b - y}{2 \cdot x}\right)^3; x = \cos(\omega \cdot t); y = \operatorname{tg}(\omega \cdot t); b = \omega^2 \cdot \sqrt{t}$	$w = 2,5; t = 3,8$
16	$y = \sqrt{x^2 + a^2} \cdot \ln(x + a); x = e^b; a = \frac{(\sin(b) + \cos(b))^2}{\pi \cdot b}$	$b = 0,45$
17	$y = e^{2x+a} \cdot \sqrt{a \cdot x^4}; x = \lg^2(a); a = \sin^2(b)$	$b = 2,92$
18	$y = x + \sqrt[3]{ a^2 + x^2 + 1}; x = a^{-1,5}; a = \sqrt{17} \cdot \frac{w}{2 \cdot \pi}$	$w = 10,5$
19	$z = 3,14 + ctg\left(\frac{x+y}{1+x \cdot y}\right); x = \lg(y \cdot t); y = e^{a \cdot t}$	$a = 3,2; t = 2$
20	$y = \frac{1 + \sin(x)}{\cos(x) \sin(x)}; x = \sqrt[3]{1 + \lg^2(a + b)}; a = e^{-b} + 1$	$b = 1,75$
21	$y = \log_3\left(\sqrt{\frac{x+a}{x \cdot a}}\right); x = (\sin(a) + \cos(a))^2; a = 1 + 0,7^{c+d}$	$c = 0,54; d = 0,13$
22	$y = \sin(x) + x \cdot e^{x+2}; x = \sqrt[4]{a^2 + b^2}; a = 2,35 \cdot \ln(b)$	$b = 0,5$
23	$y = \frac{\sin(2 \cdot x)}{a \cdot x} \cdot \sqrt{\ln(x + 2 \cdot a)}; x = e^{5 \cdot a} + \operatorname{tg}^2(a)$	$a = 0,84$
24	$y = \sin^3(0,5 \cdot c)^2; c = \sqrt{1 + (\ln^2(a) + \ln(b))^2}; a = e^{-b}$	$b = 6,5$
25	$y = \sqrt{e^{a+2} + x}; x = \frac{a+6}{c \cdot b} \cdot \sin(c \cdot b); a = \ln(c + \sin^2(b))$	$c = 1,5; b = 0,3$
26	$y = \frac{a - \sqrt[3]{b^2}}{c + a/(c + b)}; b = \sin^3(w \cdot t)^2; a = e^{c \cdot t}; c = \frac{w \cdot t}{2 \cdot \pi}$	$w = 4,1; t = 7,9$

27	$y = \ln\left(x + \lg\left(x + \frac{1}{3 \cdot x}\right)\right); \quad x = 1,2 \cdot e^{2a+0,5}; \quad a = \cos^3(b+d)$	$b = 5,2; \quad d = 2,5$
28	$y = \sin(x) + e^{a+2}; \quad x = \sqrt[3]{a^2 + b^2}; \quad a = \frac{2,35 \cdot \ln(b)}{\pi \cdot b}$	$b = 9,4$
29	$y = \lg(\ln(x+a+1)); \quad x = \sqrt[4]{b + \frac{c \cdot a}{\pi \cdot c}}; \quad a = (\sin(c) + \sin^2(b))^2$	$c = 4,65; \quad b = 3$
30	$a = y^{\sqrt[3]{ x }} + ctg^3(y-3); \quad c = \frac{tg(a - \pi/6) \cdot x}{ x +1/y^2+1}; \quad x = \lg(e^{y-2})$	$y = 5,8$

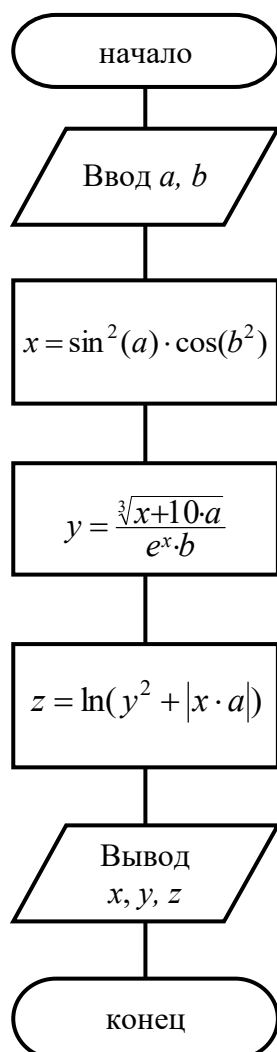
1.4. ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЯ

Разработать алгоритм и программу вычисления значений следующих функций:

$$z = \ln(y^2 + |x \cdot a|); \quad y = \frac{\sqrt[3]{x+10 \cdot a}}{e^{x \cdot b}}; \quad x = \sin^2(a) \cdot \cos(b^2).$$

Выполнить рабочий расчет для $a = 450; \quad b = 0,1$.

Блок-схема алгоритма



Паскаль-программа

```

Program example1;
var a,b,x,y,z:real;
begin
  write ('Введите a='); readln (a);
  write ('Введите b='); readln (b);
  x:=sqr(sin(a))*cos(b*b);
  y:=power((x+10*a),1/3)/(exp(x)*b);
  z:=ln(y*y+abs(x*a));
  writeln ('x=',x:1:3,' y=', y:1:3,' z=',z:1:3);
end.
  
```

Подготовка контрольного примера для тестового расчета:

Возьмем $a = 1; \quad b = 1$.

Выполним ручной счет:

$$x = \sin^2(1) \cdot \cos(1) = 0,708 \cdot 0,541 = 0,383;$$

$$y = \sqrt[3]{0,383 + 10} / e^{0,383} = 2,182 / 1,467 = 1,488;$$

$$z = \ln(1,488^2 + 0,383) = \ln(2,597) = 0,953.$$

Выполним вычисления по программе:

```

Окно вывода
Введите a=1
Введите b=1
x=0.383  y=1.488  z=0.954

```

Результаты ручного и компьютерного счета контрольного примера приблизительно совпадают, следовательно программа верна. Теперь выполним рабочий расчет с заданными исходными данными:

```

Окно вывода
Введите a=450
Введите b=0.1
x=0.467  y=103.514  z=9.299

```

1.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлении студентом аккуратно оформленного отчета и при умении разрабатывать алгоритмы и программы линейной структуры.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;
- блок-схему разработанного алгоритма решения задачи;
- текст разработанной программы;
- результаты тестового и рабочего расчета по программе;
- выводы.

1.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение алгоритма. Перечислите способы записи алгоритма.
2. Что понимается под линейной структурой алгоритма?
3. Из каких элементов состоит блок-схема алгоритма? Для чего предназначен каждый блок?
4. Что такое «язык программирования», «среда программирования»?
5. Понятие компьютерной программы и ее назначение.
6. Какова структура программы, написанной на языке Паскаль?
7. Что такое оператор (команда) языка? Из каких операторов состоит линейная программа?
8. Запишите синтаксис оператора ввода, оператора вывода, оператора присваивания.
9. Что такое «переменная»?

10. Что такое «тип данных»? Для чего нужен тип данных? Какие типы данных есть в Паскале?
11. Для чего нужен раздел объявления переменных? Как объявляются переменные?
12. Как записываются арифметические операции в языке Паскаль?
13. Какие встроенные математические функции есть в Паскале?

2. РАБОТА №2. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЕННОЙ СТРУКТУРЫ

2.1. ЦЕЛЬ РАБОТЫ

Формирование у студентов умения анализировать исходные данные задачи, получение навыков разработки и программирования вычислительного процесса разветвленной структуры.

2.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

Разветвленным называется алгоритм, который предусматривает выбор одной из нескольких возможных последовательностей действий (ветвей) в зависимости от некоторых условий. Разновидностями ветвления являются структуры «если – то» и «если – то – иначе», показанные на рис. 1.

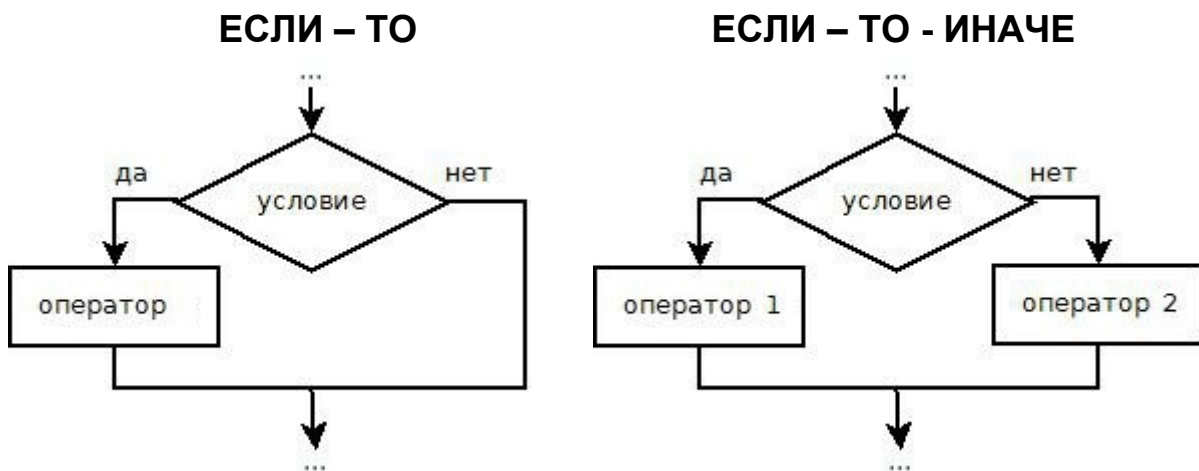


Рис. 1. Разновидности разветвленной структуры

В программе на языке Паскаль для организации разветвления используют условный оператор и оператор выбора.

Формат условного оператора:

If <Условие> **Then** <Оператор 1> **Else** <Оператор 2>;

Выполняется условный оператор следующим образом: вычисляется значение условия, если оно – “истина” (*true*), то выполняется оператор 1, а оператор 2 пропускается, и наоборот, если значение условия – “ложь” (*false*), то оператор 1 пропускается, а оператор 2 выполняется.

Также возможно использование короткой формы условного оператора:

If <Условие> **Then** <Оператор>;

Здесь, если условие – “истина”, то выполняется оператор, а если – “ложь”, то оператор пропускается.

Формат оператора выбора:

Case <Ключ> **Of** <Список выбора> **Else** <Оператор> **End**;

Здесь <Список выбора> представляет собой несколько инструкций вида: <Константа>:<Оператор>. Оператор выбора работает следующим образом. Вычисляется значение ключа, а затем в списке выбора отыскивается константа, равная значению ключа. Если такая будет найдена, то выполняется стоящий после нее оператор, в противном случае выполняется оператор, стоящий за *Else*.

2.3. ЗАДАНИЯ К РАБОТЕ

Разработать алгоритм и программу вычисления значения заданной кусочно-непрерывной функции для произвольных значений исходных данных. Составить два варианта алгоритма (программы): а) ветвление с помощью условных операторов; б) ветвление с использованием оператора выбора.

Варианты заданий

№ варианта	Заданные функции	Исходные данные
1	$y = \begin{cases} z^2 + c, & z > c \\ \sqrt{z} + \sqrt{c}, & z < c \\ e^z + \ln c, & z = c \end{cases}, \text{ где } \begin{cases} z = \cos(\omega \cdot t) - 1; \\ c = \ln \omega \cdot t \end{cases}$	$w = 2,25; t = 3,7$
2	$y = \begin{cases} \sqrt{y+b}, & xb \leq 0 \\ \sqrt[3]{y-b}, & 0 < xb < 1,3 \\ \sqrt[5]{b-y}, & xb > 1,3 \end{cases}, \text{ где } x = \sin(e^b)$	$b = 2,75$
3	$y = \begin{cases} 0, & ax \leq 1 \\ e^{x-a}, & 1 < ax < 5 \\ \lg(a-x), & ax \geq 5 \end{cases}, \text{ где } x = \sin a $	$a = 10,25$
4	$y = \begin{cases} \lg(a/x), & ax \leq -1 \\ e^{x/a}, & -1 < ax < 1 \\ x-a , & ax \geq 1 \end{cases}, \text{ где } x = \sqrt{a^2 + 3}$	$a = 1,45$

5	$y = \begin{cases} 1,178, & x \leq 0 \\ \sqrt{x + \ln(a)}, & 0 \leq x \leq a \\ e^{x-a} + 1, & x > a \end{cases}$	, где $x = 5,5 \cdot \sin a$	$a = 6,5$
6	$y = \begin{cases} a + 2/b + 4, & a > b \\ (a+b)^2, & b = a \\ b, & a < b \end{cases}$	, где $a = \cos(w \cdot t);$ $b = \sin(w \cdot t)$	$w = 0,5; t = 3,2$
7	$y = \begin{cases} \lg 2,5, & x \leq 0 \\ 1 + \frac{\ln(x+2)}{x+2}, & 0 < x \leq 2 \\ e^{0,3x} - 2x, & x > 2 \end{cases}$	, где $x = a \ln(a+b)$	$a = 4,5; b = 1,36$
8	$y = \begin{cases} \ln(x^2 + 7), & x < 0 \\ 1,35 + \sqrt{x^2 + 2}, & 0 \leq x \leq 7 \\ 0,35x^2, & x > 7 \end{cases}$	, где $x = \ln a + \ln b$	$a = 2,45; b = 0,37$
9	$y = \begin{cases} a^2 - x, & x \leq 0 \\ \frac{a-x}{\ln(x+a)}, & 0 < x \leq a \\ \sqrt{a^2 - x}, & x > a \end{cases}$	, где $x = \ln(3,5 + a)$	$a = 2,2$
10	$y = \begin{cases} \ln^2(a+x), & x \leq 0 \\ \frac{\cos(a \cdot x)}{a+x}, & 0 < x < 5.4 \\ \sqrt{\ln x + a^2}, & x \geq 5.4 \end{cases}$	, где $x = \ln(1.7 + a^{0.3})$	$a = 1,8$
11	$y = \begin{cases} \lg 2,5, & x \leq 0 \\ 1 + \frac{\ln(x+2)}{x+2}, & 0 < x \leq 2 \\ e^{0,3x} - 2x, & x > 2 \end{cases}$	, где $x = 3 \cdot \ln(a+b)$	$a = 5,15; b = 1,73$
12	$y = \begin{cases} \ln(x+z), & x < 0 \\ 1,35 + \sqrt{x+z}, & 0 \leq x \leq z \\ 0,35x, & x > z \end{cases}$	, где $x = \ln a - \ln b,$ $z = \sqrt{a+b}$	$a = 1,64; b = 0,73$
13	$y = \begin{cases} 1, & ax \leq 0 \\ e^{(0,35a+x)} + ax, & 0 < ax \leq 5 \\ \ln(a \cdot x) + \sin x, & ax > 5 \end{cases}$	, где $x = \sin a + \ln b$	$a = 7,2; b = 0,59$

14	$y = \begin{cases} 0, & x = 0 \\ x, & x = y \\ 2^{-x^2/9 - y^2/9}, & x \neq y \end{cases}$	, где $\begin{aligned} x &= \cos(w^2 \cdot t) , \\ y &= \sin(w \cdot t^2) \end{aligned}$	$w = 3,5; \quad t = 6,8$
15	$y = \begin{cases} 16x + 9x^2 - 2, & 1 \leq x \leq 9 \\ 0, & x < 1 \\ a - x, & x \geq 9 \end{cases}$	, где $x = \ln(\operatorname{tg} a)$	$a = 4,38$
16	$y = \begin{cases} \lg 5, & x \leq 0 \\ e^{x-2}, & 0 < x < 1,3 \\ \operatorname{tg}^{xa}, & x \geq 1,3 \end{cases}$	, где $x = a \ln(a \cdot b)$	$a = 3,2; \quad b = 1,35$
17	$y = \begin{cases} \sin^2 x, & x \leq 0 \\ \lg x / a, & 0 < x < \pi \\ x^2 / a + \ln a, & x \geq \pi \end{cases}$	, где $x = e^{a/b} a^3$	$a = 2,9; \quad b = 1,75$
18	$y = \begin{cases} \sin(a+b), & (a+b) \leq 0 \\ \sqrt{a^2 + b^2}, & 0 < (a+b) < 7 \\ a - b, & (a+b) \geq 7 \end{cases}$	, где $b = \frac{\sin a}{\cos(2+a)}$	$a = 3,37$
19	$y = \begin{cases} 10^{x-a}, & ax > 0 \\ e^{x+a}, & ax = 0 \\ \operatorname{tg}(x+a), & ax < 0 \end{cases}$	, где $x = \cos a$	$a = 2,55$
20	$y = \begin{cases} 0, & x \leq 0 \\ \lg \ln x , & 0 < x < 10 \\ \sqrt{\sin a^2 x}, & x \geq 10 \end{cases}$	, где $x = \frac{\operatorname{tga}}{\sqrt{a}}$	$a = 3,5$
21	$y = \begin{cases} e^{x/a}, & ax \leq 0 \\ \ln ax, & 0 < ax < 1 \\ \ln(a + e^{ax}), & ax > 1 \end{cases}$	, где $x = \sin \frac{a}{2}$	$a = 6,12$
22	$y = \begin{cases} 1 - \sin x, & 0 \leq x \leq 0,1 \\ 0,5(1 + \cos x), & 0,1 < x \leq 0,6 \\ 0, & \text{в ост. случаях} \end{cases}$	, где $x = \cos(w \cdot t)$	$w = 0,25; \quad t = 0,92$
23	$y = \begin{cases} a\sqrt{1 - x^2/b^2}, & 0 < x < b \\ -\sqrt{ x - b}, & b \leq x \\ a/b(x+b), & -b < x \leq 0 \end{cases}$	, где $\begin{aligned} a &= \cos(w^2 \cdot t); \\ b &= \sin(w \cdot t^2); \\ x &= t \cdot \ln w \end{aligned}$	$w = 1,5; \quad t = 0,15$

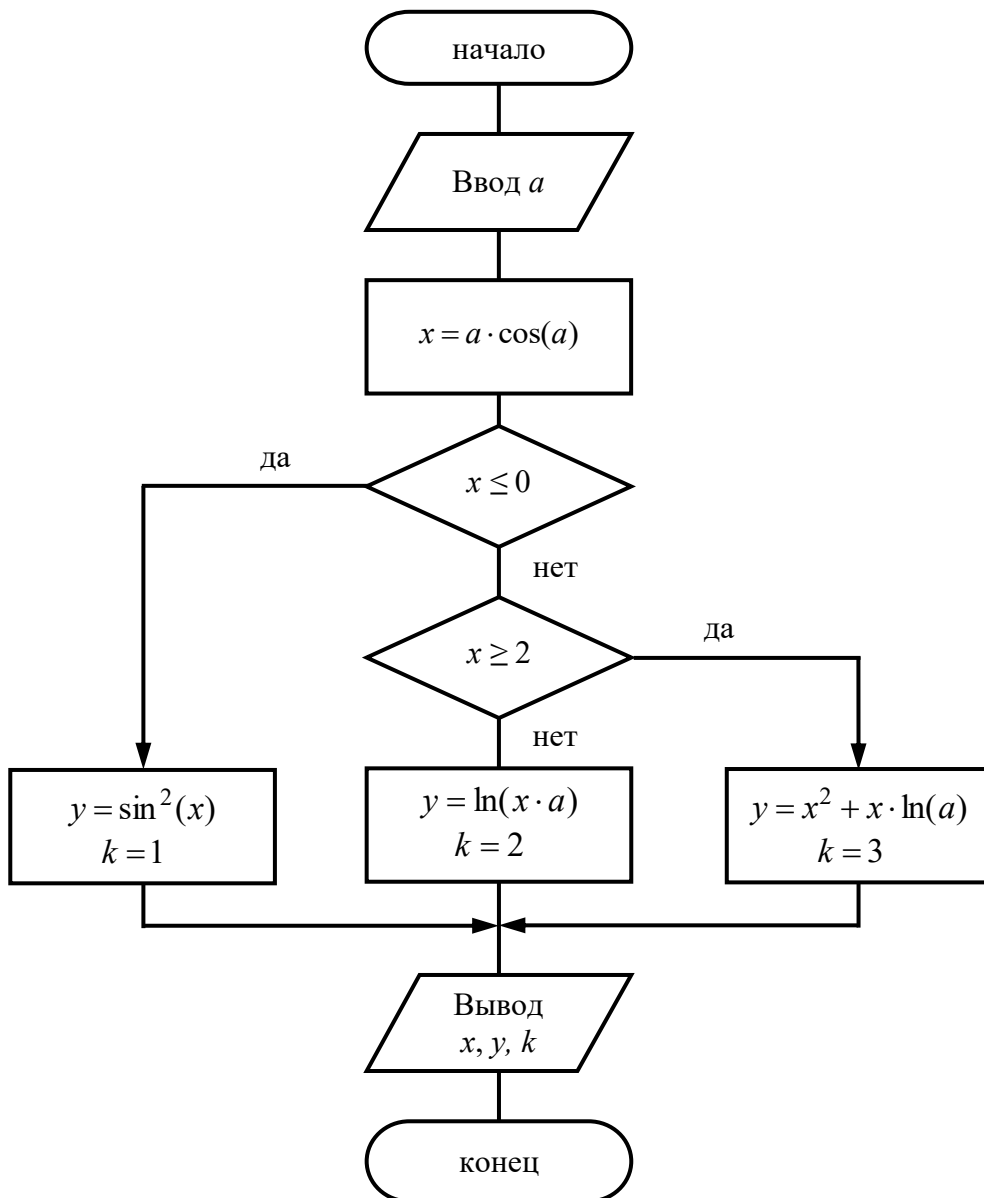
24	$y = \begin{cases} a, & x \leq 0 \\ 2a + x^{0,5a}, & 0 < x < 2,5 \\ (x+a)^{0,7} + a^3, & x \geq 2,5 \end{cases}, \text{ где } x = 3,43 - e^{-b}$	$b = 2,5$
25	$y = \begin{cases} \sin(a+b), & (a+b) \leq 0 \\ \sqrt{a^2 + b^2}, & 0 < (a+b) < 2,15 \\ 3,437, & \text{в ост. случаях} \end{cases}, \text{ где } b = \cos(2+a)$	$a = 2,5$
26	$y = \begin{cases} -4, & x < 0 \\ x^2 + 3x + 4, & 0 \leq x < 1 \\ (x^2 + 3x + 4)^2, & x \geq 1 \end{cases}, \text{ где } x = \operatorname{tg}(w \cdot t)$	$w = 3,1; t = 6,3$
27	$y = \begin{cases} 0, & x \leq 0 \\ \sin x + \cos(ax), & 0 < x \leq 2 \\ \sqrt{\sin^2 x + \cos^2 ax}, & x > 2 \end{cases}, \text{ где } x = \sqrt[3]{a+2}$	$a = 3,35$
28	$y = \begin{cases} 3,35, & x \leq 0 \\ (a+x)e^{a+x}, & 0 < x < 1 \\ \sqrt{ax+1/(a+x)}, & x > 1 \end{cases}, \text{ где } x = \sin(a \cdot b)$	$a = 3,2; b = 2,7$
29	$y = \begin{cases} (a+x)^2, & x \leq 0 \\ \frac{a+x}{\sin x + ax}, & 0 < x < 2.2 \\ \sqrt{a^2 + x^2}, & x \geq 2.2 \end{cases}, \text{ где } x = 5 \cos(a \cdot b)$	$a = 1,6; b = 0,83$
30	$y = \begin{cases} x + 2 \operatorname{tg} x, & x \geq 1 \\ \frac{1}{1+x}, & x \leq 0 \\ 2x + \ln x, & 0 \leq x \leq 1 \end{cases}, \text{ где } x = a^2 - b^2 - a \cdot c$	$a = 2,56; b = 0,73; c = 1,52$

2.4. ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЯ

Разработать алгоритм и программу вычисления значения следующей кусочно-непрерывной функции:

$$y = \begin{cases} \sin^2 x & \text{если } x \leq 0 \\ \ln(x \cdot a) & \text{если } 0 < x < 2 \\ x^2 + x \cdot \ln a & \text{если } x \geq 2 \end{cases}, \text{ где } x = a \cdot \cos(a).$$

Выполнить рабочий расчет для $a = 5,3$.

Блок-схема алгоритма**Паскаль-программа**

```

Program example2;
var a,x,y:real; k:integer;
begin
  write ('Введите a='); readln (a);
  x:=a*cos(a);
  if x<=0 then
    begin y:=sqr(sin(x)); k:=1; end

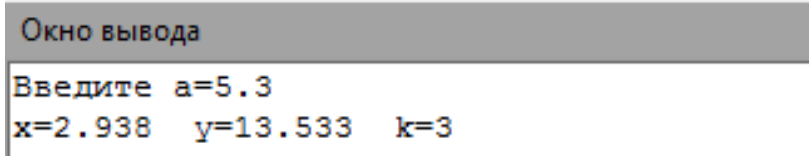
```

```

else
  if x>=2 then
    begin y:=sqr(x)+x*ln(a); k:=3; end
  else
    begin y:=ln(x*a); k:=2; end;
writeln ('x=',x:1:3,' y=', y:1:3,' k=',k:1);
end.

```

Выполним рабочий расчет с заданным значением исходного данного:



Окно вывода

Введите a=5.3

x=2.938 y=13.533 k=3

2.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлении студентом аккуратно оформленного отчета и при умении разрабатывать алгоритмы и программы разветвленной структуры.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;
- блок-схему разработанного алгоритма решения задачи;
- текст разработанной программы;
- результаты тестового и рабочего расчета по программе;
- выводы.

2.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что понимается под разветвленной структурой алгоритма?
2. Какие есть разновидности «ветвления»? Примеры графических схем с ветвлением?
3. Какие операторы языка Паскаль позволяют создавать программы с ветвлением?
4. Синтаксис полной формы условного оператора. Выполнение условного оператора в программе.
5. В каком случае необходимо заключать операторы в операторные скобки begin ... end?
6. Синтаксис короткой формы условного оператора и его выполнение в программе.
7. Синтаксис оператора выбора и его выполнение в программе.

3. РАБОТА №3. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ

3.1. ЦЕЛЬ РАБОТЫ

Формирование у студентов умения анализировать исходные данные задачи, изучение различных видов циклических алгоритмов, получение навыков разработки и программирования вычислительного процесса циклической структуры.

3.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

Циклом называется алгоритм, предусматривающий многократное повторение однотипных действий. «Телом цикла» называют те действия, которые нужно многократно повторять.

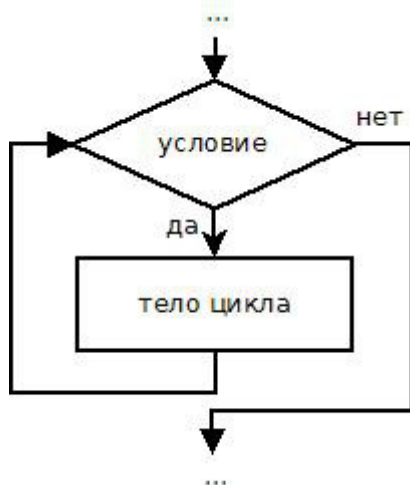
Различают циклы с заранее известным числом повторений (цикл с параметром)



Рис. 2. Структура цикла с параметром

и итерационные циклы (циклы с предусловием и с постусловием), в которых число повторений заранее неизвестно и зависит от управляющего условия (рис. 3).

ЦИКЛ С ПРЕДУСЛОВИЕМ



ЦИКЛ С ПОСТУСЛОВИЕМ

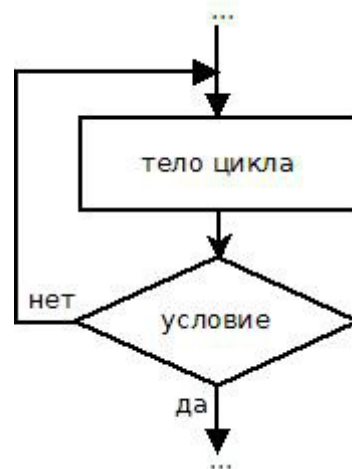


Рис. 3. Структуры итерационных циклов

Для трех видов цикла существуют соответствующие операторы в языке Паскаль.

Оператор цикла «Пока» или цикла с предусловием записывается следующим образом:

While <Условие> **Do** <Оператор>;

Словами можно прочесть так: "Пока истинно условие, повторять оператор". Выход из цикла произойдет, когда условие станет ложным.

Формат оператора цикла «До» или цикла с постусловием:

Repeat <Оператор 1>; <Оператор 2>; . . . <Оператор n>; **Until** <Условие>;

Читается так: "Выполнять операторы тела цикла до выполнения условия". Этот вид цикла отличается от предыдущего тем, что операторы тела хотя бы один раз выполняются.

Цикл «С параметром» создается с помощью специальной целочисленной переменной - параметра цикла, которая будет изменяться на единицу при каждом повторении тела цикла. Таким образом, задав начальное и конечное значения для такой переменной, можно точно установить количество выполнений тела цикла. У этого вида цикла предусмотрено два формата:

For <И.П.>:=<Н.З.> **To** <К.З.> **Do** <Оператор>;

For <И.П.>:=<Н.З.> **Downto** <К.З.> **Do** <Оператор>;

Здесь И.П. - имя переменной-параметра, Н.З. - его начальное значение, К.З. - соответственно конечное значение параметра. В качестве начального и конечного значений можно применять произвольные выражения целого типа.

Читается данная структура так: "Для параметра, изменяющегося от начального значения до конечного повторять оператор (тело цикла)". В первом формате параметр с каждым повторением увеличивается на единицу, а во втором - уменьшается.

3.3. ЗАДАНИЯ К РАБОТЕ

Разработать алгоритм и программу вычисления таблицы значений двух заданных функций для произвольного диапазона изменения независимого аргумента, определяемого начальным $x_{нач}$ и конечным $x_{кон}$ значениями, а также шагом его изменения Δx . Составить три варианта алгоритма (программы): а) с использованием оператора цикла с предусловием; б) с использованием оператора цикла с постусловием; в) с использованием оператора цикла с параметром.

Варианты заданий

№ варианта	Заданные функции	Диапазон изменения аргумента		
		$x_{нач}$	$x_{кон}$	Δx
1	$y = \sqrt[3]{x \cdot (2,7 \cdot x^{1,35} + 1,2 \cdot g)}$; $g = tg^2(\frac{x}{2\pi})$	1,2	2	0,2
2	$z = 0,87 \cdot x^y - \frac{y}{3 \cdot x}$; $y = (x + 2,3 \cdot e^{\pi+x}) \cdot \sqrt{\pi \cdot x}$	3,5	5,5	0,5

3	$y = \cos^2(\frac{x}{3u}); \quad u = \lg(x) \cdot (2,35 \cdot (x+1)^x + 2 \cdot x)$	1,5	3,5	0,5
4	$y = \frac{x^2 \cdot a^3 + x^{0,8} \cdot a}{\sin(x^2) + \sin(a)}; \quad a = 3,7 \cdot \sqrt{e^{-x}}$	2,2	2,3	0,25
5	$z = e^y + \log_2^3(x); \quad y = \frac{\ln(2 \cdot x + 3,47)}{\sqrt{(x+1)^3}}$	0,5	2	0,6
6	$y = \frac{x^2 - 3 \cdot p + 2}{\sqrt{2 \cdot x^3 - \ln(x)}}; \quad p = \sin(x) + 2$	5	10	1
7	$y = \frac{a \cdot x}{(1 + (a-1) \cdot x) \cdot \ln(x+1)}; \quad a = 3,48 \cdot \frac{e^x + e^{-x}}{2 \cdot \pi}$	0,3	3	0,3
8	$z = \frac{\sqrt{2 \cdot y + 1}}{2 \cdot x}; \quad y = \frac{1}{x \cdot (1,35 \cdot x^{0,7} + \sqrt{2 \cdot x + 1})}$	1	5	1
9	$f = \frac{a+x}{2 \cdot a}; \quad a = (x + 2,3 \cdot e^{2,5+x}) \cdot \sqrt[3]{3 \cdot x}$	3,5	7,5	1
10	$y = \frac{x^3 - a \cdot x^2 + \sqrt[4]{x}}{(x^2 + a \cdot x)^{0,3}}; \quad a = 3,34 \cdot tg^3(x) $	2	4	0,5
11	$y = \frac{5,1 \cdot (a - 0,1 \cdot x^2)}{e^x \cdot (a+x)}; \quad a = \sin^3(\frac{2,5 \cdot x}{2 \cdot \pi})$	1	3	0,5
12	$y = \frac{z + \sqrt{x + 0,7 \cdot x^2 + 5,3}}{2 \cdot e^{0,7 \cdot x}}; \quad z = \log_2(x)$	1	5	1
13	$y = \frac{\sin^2(x+0,42)}{\sqrt[2]{t+4,75}}; \quad t = \frac{\cos^2(x+0,42)}{\sqrt[2]{x+4,78}}$	0,1	1,1	0,2
14	$y = tg^3(\frac{u}{x \cdot 2,4}); \quad u = \frac{1,5 + 2,7 \cdot \ln(5+x)}{1 + 1/x}$	3	5	0,5
15	$w = ctg^3(3,4 \cdot y); \quad y = \frac{2,5 \cdot x + 3 \cdot x^{0,7}}{\sqrt{x+1}}$	2	8	2
16	$z = y \cdot e^x + \ln(\frac{x+x^2-2 \cdot x+1}{e^x}); \quad y = e^{-x} + \frac{x}{2 \cdot \pi}$	1	5	0,5
17	$y = x^{0,6} + 2 \cdot z + \lg(x+1); \quad z = tg^2(x + \frac{1}{2 \cdot x})$	1	2,5	0,5
18	$z = \ln(y) - \pi \cdot x; \quad y = \sqrt[3]{x^4 + 2 \cdot (x+1)}$	2	7	1
19	$b = \sqrt{(x+1) \cdot (3,5 \cdot x^{2,5} + a)}; \quad a = tg^2(0,2 \cdot x)$	1	4	0,5

20	$y = x^3 + \frac{2,5 \cdot x}{\ln(t+1)}; \quad t = 2,1 \cdot \sin^2(x^3)$	1	5	1
21	$y = \ln(x^2) - tg(ctg(g)); \quad g = \frac{x+2,5}{5 \cdot x}$	1	3	0,25
22	$z = 3,5^x - e^x + ctg(y); \quad y = \frac{x}{\pi \cdot \lg(x^2)}$	-4	-1	0,5
23	$y = \ln(\ln(a)) + \sin(2 \cdot a); \quad a = \frac{x^2 + \pi}{3 \cdot x}$	1	5	0,5
24	$y = \frac{c}{(1+x) \cdot \pi}; \quad c = \ln(x) ^2 - \sin(x \cdot tg(x))$	0,1	0,2	0,01
25	$z = \ln a + ctg^2(a); \quad a = e^x - x + \sin(x)$	-8	-5	0,5
26	$y = \sin(\sqrt{x}) + 0,5^z; \quad z = \sqrt[3]{x^2 + 3,5}$	2	6	0,5
27	$z = \sin(\sqrt{x-3,2}) + b^x; \quad b = \sqrt[3]{\frac{10}{\pi \cdot x}}$	5,2	7,2	0,4
28	$y = \pi^x \cdot \sqrt{\ln(c)}; \quad c = 1 + \left \frac{x+3}{5 \cdot x} \right $	2	12	1,5
29	$a = \ln y + \sqrt[3]{y^2}; \quad y = \frac{\sin(x)}{\ln(x)} + ctg(x) $	3	6	0,5
30	$y = \frac{x+x^3 \cdot a^{0,7}}{\sin(x^2) + \sin(a)}; \quad a = \log_3(x) + \sin^3(x^2)$	2	3	0,2

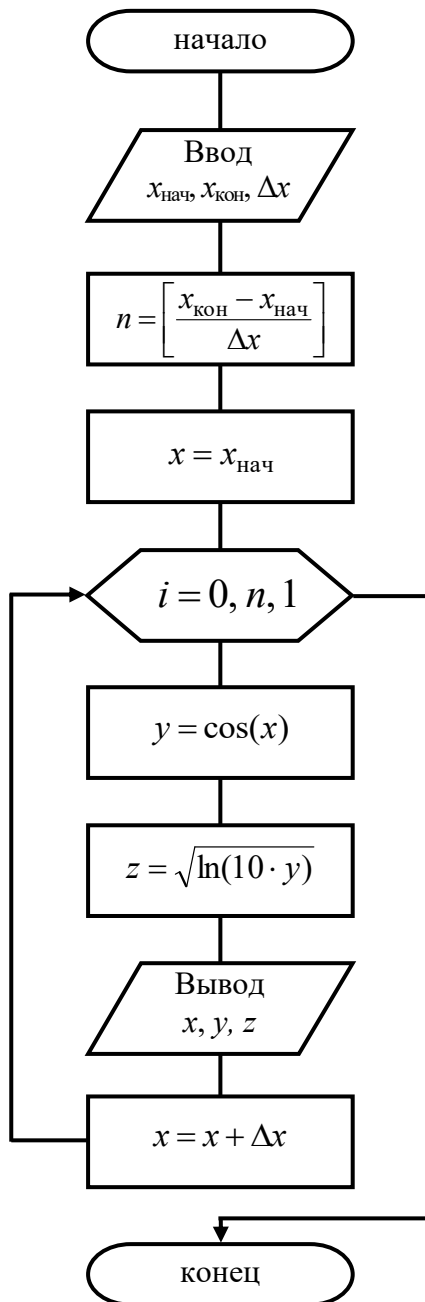
3.4. ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЯ

Разработать алгоритм и программу вычисления таблицы значений следующих функций:

$$z = \sqrt{\ln(10 \cdot y)}; \quad y = \cos(x)$$

для значений аргумента x , изменяющегося от $x_{\text{нач}}=0,5$ до $x_{\text{кон}}=1,2$ с шагом $\Delta x=0,1$.

Для решения поставленной задачи применим циклический алгоритм с изменением параметра (с заданным числом повторений). Соответственно в программе используем оператор **For**.

Блок-схема алгоритмаПаскаль-программа

```

Program example3;
var x,y,z,xn,xk,dx:real; i,n:integer;
begin
  write ('Введите xn, xk, dx=');
  readln (xn, xk, dx);
  n:=trunc((xk-xn)/dx);
  x:=xn;
  for i:=0 to n do
  begin
    y:=cos(x);
    z:=sqrt(ln(10*y));
    writeln ('x=',x:1:2,' y=', y:1:3,' z=',z:1:3);
    x:=x+dx;
  end;
end.

```

Выполним рабочий расчет с заданными значениями исходных данных:

Окно вывода		
Введите xn, xk, dx=0.5 1.2 0.1		
x=0.50	y=0.878	z=1.4740
x=0.60	y=0.825	z=1.4531
x=0.70	y=0.765	z=1.4262
x=0.80	y=0.697	z=1.3933
x=0.90	y=0.622	z=1.3524
x=1.00	y=0.540	z=1.2995
x=1.10	y=0.454	z=1.2306
x=1.20	y=0.362	z=1.1357

3.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлении студентом аккуратно оформленного отчета и при умении разрабатывать алгоритмы и программы циклической структуры.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;

- блок-схему разработанного алгоритма решения задачи;
- текст разработанной программы;
- результаты тестового и рабочего расчета по программе;
- выводы.

3.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что понимается под циклической структурой алгоритма?
2. Приведите примеры задач, для решения которых используется структура «цикл».
3. Какие есть разновидности циклических структур? Примеры графических схем с циклами?
4. Что такое «тело цикла»?
5. Какие операторы языка Паскаль позволяют создавать циклические программы?
6. Синтаксис оператора цикла с предусловием. Как он выполняется в программе?
7. Что требуется сделать для того, чтобы в теле цикла выполнялись несколько операторов?
8. Синтаксис оператора цикла с постусловием. Как он выполняется в программе?
9. Запишите синтаксис оператора цикла с параметром. Его выполнение в программе.
10. Что такое «параметр цикла»? С каким шагом перебираются значения параметра цикла? От чего это зависит?

4. РАБОТА №4. ПРОГРАММИРОВАНИЕ ИТЕРАЦИОННЫХ ЦИКЛОВ. СУММИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ РЯДОВ

4.1. ЦЕЛЬ РАБОТЫ

Формирование у студентов умения анализировать исходные данные задачи, изучение итерационных циклических алгоритмов на примере вычисления суммы функционального сходящегося ряда, получение навыков разработки и программирования алгоритмов с вложенными циклами.

4.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

Итерационным называют такой цикл, в котором количество повторений тела цикла до выполнения алгоритма неизвестно. Выход из такого цикла происходит при достижении определённой цели («срабатывания» условия выхода из цикла).

Типичным примером использования итерационного цикла является алгоритм приближенного вычисления функции через ее представление в виде бесконечного сходящегося функционального (степенного) ряда. Например:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Функция представляется в виде суммы слагаемых (членов ряда), каждый из которых может быть вычислен по формуле общего члена ряда:

$$y = \sin(x) = \sum_{i=0}^{\infty} \frac{(-1)^i \cdot x^{2i+1}}{(2i+1)!}.$$

Если просуммировать только часть слагаемых, то функция будет вычислена приближенно. Чем больше членов ряда суммируем, тем точнее вычисляем значение функции. Таким образом, аналитическое вычисление функции y можно заменить вычислением с помощью цикла суммы членов ряда a_i :

$$S = b + a_0 + a_1 + \dots + a_i + \dots,$$

где i – порядковый номер члена, b – постоянный начальный член ряда.

Важно то, что каждый последующий (по номеру) член может быть вычислен через предыдущий по рекуррентной формуле:

$$a_i = R \cdot a_{i-1}, \quad i = 1, 2, \dots,$$

где R – рекуррентный множитель (его формула выводится из формулы общего члена ряда).

Из теории известно, что абсолютная погрешность $eps = |y - S|$ приближенного вычисления функции разложением её в ряд будет зависеть от количества поочередно вычисляемых и включаемых в сумму членов ряда. И эта погрешность по абсолютной величине не будет превышать первого из отбрасываемых членов ряда. Следовательно, можно построить алгоритм суммирования сходящегося ряда в виде итерационного цикла с условием выхода из него по достижению требуемой точности вычисления функции.

4.3. ЗАДАНИЯ К РАБОТЕ

Функция задана двумя способами: аналитической формулой $y = f(x)$ и ее разложением в бесконечный ряд S .

Разработать алгоритм и программу вычисления точного y и приближенного S значений функции при изменении её аргумента от начального значения $x_{\text{нач}}$ до конечного $x_{\text{кон}}$ с шагом Δx . Приближенное значение вычислить путем суммирования членов сходящегося функционального ряда до достижения требуемой точности ε . Предусмотреть завершение процесса суммирования членов ряда по заданному максимальному номеру члена ряда n для предотвращения закливания итерационного цикла. Результаты вычислений вывести в виде таблицы.

Варианты заданий

№ варианта	Заданная функция и ее разложение в ряд	Диапазон изменения аргумента		
		$x_{нач}$	$x_{кон}$	Δx
1	$y = \ln(1+x); \quad S = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{i+1} \frac{x^i}{i} + \dots$	0,1	0,6	0,1
2	$y = e^{2 \cdot x} - x; \quad S = 1 + x + \frac{(2 \cdot x)^2}{2!} + \frac{(2 \cdot x)^3}{3!} + \dots + \frac{(2 \cdot x)^i}{i!} + \dots$	0,1	0,8	0,1
3	$y = 3^x + 3; \quad S = 4 + x \cdot \ln(3) + x^2 \cdot \frac{\ln^2(3)}{2} + \dots + x^i \cdot \frac{\ln^i(3)}{i!} + \dots$	0,2	1,1	0,2
4	$y = x - \sin^2(x); \quad S = x - x^2 + \frac{2^3}{4!} \cdot x^4 + \dots + (-1)^i \cdot \frac{2^{2 \cdot i - 1}}{(2 \cdot i)!} \cdot x^{2 \cdot i} + \dots$	0,5	0,9	0,1
5	$y = e^x - 3 \cdot x; \quad S = 1 - 2 \cdot x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^i}{i!} + \dots$	0,1	0,7	0,1
6	$y = e^{-4 \cdot x^2}; \quad S = 1 - (2 \cdot x)^2 + \frac{(2 \cdot x)^4}{2!} - \dots + (-1)^i \cdot \frac{(2 \cdot x)^{2 \cdot i}}{i!} + \dots$	-1	-0,6	0,05
7	$y = x + \sin(x); \quad S = 2 \cdot x - \frac{x^3}{3!} + \dots + (-1)^i \cdot \frac{x^{2 \cdot i + 1}}{(2 \cdot i + 1)!} + \dots$	-0,9	-0,5	0,1
8	$y = 2 \cdot (\cos^2(x) - 1); \quad S = -\frac{(2 \cdot x)^2}{2} + \dots + (-1)^i \cdot \frac{(2 \cdot x)^{2 \cdot i}}{(2 \cdot i)!} + \dots$	0	0,75	0,15
9	$y = \frac{\sin(x)}{x}; \quad S = 1 - \frac{x^2}{3!} + \dots + (-1)^i \cdot \frac{x^{2 \cdot i}}{(2 \cdot i + 1)!} + \dots$	0,1	0,5	0,1
10	$y = \frac{1}{1+x} + \ln(1+x); \quad S = 2 \cdot x - \frac{3 \cdot x^2}{2} \dots + (-1)^{i+1} \cdot \frac{(i+1) \cdot x^i}{i} + \dots$	0,2	0,6	0,1
11	$y = \cos(x) + x \cdot \sin(x); \quad S = 1 + \frac{x^2}{2} + \dots + (-1)^{i-1} \cdot \frac{2 \cdot i - 1}{(2 \cdot i)!} \cdot x^{2 \cdot i} + \dots$	0	0,75	0,15
12	$y = 2^x + x \cdot \ln(2) - 1; \quad S = x \cdot \ln(2) + \frac{\ln^2(2)}{n!} \cdot x^2 + \dots + \frac{\ln^i(2)}{i!} \cdot x^i + \dots$	0,1	0,5	0,1
13	$y = \cos(x) + \sin(x) - x; \quad S = \cos(x) - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^i \cdot \frac{x^{2 \cdot i + 1}}{(2 \cdot i + 1)!} + \dots$	-0,8	-0,2	0,2
14	$y = e^x \cdot \sin(x) + 1; \quad S = 1 + \sqrt{2} \cdot \sin\left(\frac{\pi}{4}\right) \cdot x + \dots + \frac{\sqrt{2} \cdot \sin\left(\frac{i \cdot \pi}{4}\right)}{i!} \cdot x^i + \dots$	0,2	0,7	0,1
15	$y = \frac{e^x - e^{-x}}{2}; \quad S = x + \frac{x^3}{3!} + \dots + \frac{x^{2 \cdot i + 1}}{(2 \cdot i + 1)!} + \dots$	1	2	0,1
16	$y = \frac{4}{3} \cdot \sin^3(x) - 1; \quad S = -1 + \frac{8}{3!} \cdot x^3 - \dots + (-1)^{i+1} \cdot \frac{3^{2 \cdot i - 1}}{(2 \cdot i + 1)!} \cdot x^{2 \cdot i + 1} + \dots$	0,1	0,5	0,1
17	$y = 2 \cdot \cos^2(x) - 1,8; \quad S = 0,2 - \frac{(2 \cdot x)^2}{2!} + \frac{(2 \cdot x)^4}{4!} - \dots + (-1)^i \cdot \frac{(2 \cdot x)^{2 \cdot i}}{(2 \cdot i)!} + \dots$	0,2	0,8	0,1

18	$y = (\frac{1}{x} + x + x^2) \cdot e^{-x}; \quad S = \frac{1}{x} - \frac{x}{2} + \frac{2^3}{3!} \cdot x^2 - \dots + (-1)^i \cdot \frac{i^3}{(i+1)!} \cdot x^i + \dots$	2	5	0,5
19	$y = x - \sin(x); \quad S = \frac{x^3}{3!} - \frac{x^5}{5!} + \frac{x^7}{7!} - \dots + (-1)^{i-1} \cdot \frac{2^{2 \cdot i - 1}}{(2 \cdot i - 1)!} + \dots$	0	2	0,5
20	$y = \sin(2 \cdot x) - 2 \cdot x; \quad S = -\frac{(2 \cdot x)^3}{3!} + \dots + (-1)^i \cdot \frac{(2 \cdot x)^{2 \cdot i + 1}}{(2 \cdot i + 1)!} + \dots$	0,1	0,9	0,2
21	$y = 1 + e^{-x^2}; \quad S = 2 - x^2 + \dots + (-1) \cdot \frac{x^{2 \cdot i}}{i!} + \dots$	-3	-1	0,5
22	$y = 1 + \cos(x); \quad S = 2 - \frac{x^2}{2} + \dots + (-1)^i \cdot \frac{x^{2 \cdot i}}{(2 \cdot i)!} + \dots$	0,1	0,9	0,2
23	$y = e^x \cdot \cos(x) + 2; \quad S = 3 + \sqrt{2} \cdot \cos(\frac{\pi}{4}) \cdot x + \dots + \frac{(\sqrt{2})^i \cdot \cos(i \cdot \pi/4)}{i!} \cdot x^i + \dots$	0,1	0,7	0,2
24	$y = (1 + x) \cdot e^{-x}; \quad S = 2 - \frac{x^2}{2} + \dots + (-1)^{i+1} \cdot \frac{i-1}{i!} \cdot x^i + \dots$	-0,9	-0,4	0,1
25	$y = \sin(3 \cdot x); \quad S = 3 \cdot x - \frac{(3 \cdot x)^3}{3!} + \frac{(3 \cdot x)^5}{5!} - \dots + (-1)^i \cdot \frac{(3 \cdot x)^{2 \cdot i + 1}}{(2 \cdot i + 1)!} + \dots$	0	3	0,5
26	$y = \frac{x}{2} \cdot (x \cdot \cos(x) + \sin(x)); \quad S = \cos(x) - 1 + \frac{3}{2!} \cdot x^2 + \dots + (-1)^{i+1} \cdot \frac{2 \cdot i + 1}{(2 \cdot i)!} \cdot x^{2 \cdot i} + \dots$	0,1	0,5	0,1
27	$y = (1 + 2 \cdot x^2) \cdot e^{x^2}; \quad S = 1 + 3 \cdot x^2 + \frac{5}{2} \cdot x^4 + \dots + \frac{2 \cdot i + 1}{i!} \cdot x^{2 \cdot i} + \dots$	0,1	0,7	0,2
28	$y = 1 - \cos(x); \quad S = \frac{x^2}{2!} - \frac{x^4}{4!} + \frac{x^6}{6!} - \dots + (-1)^{i+1} \cdot \frac{x^{2 \cdot i}}{(2 \cdot i)!} + \dots$	0	0,75	0,15
29	$y = \arctg(x) - x; \quad S = -\frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^i \cdot \frac{x^{2 \cdot i + 1}}{2 \cdot i + 1} + \dots$	0,1	0,7	0,2
30	$y = e^{-x^2/2}; \quad S = 1 - \frac{x^2}{2} + \frac{(x^2/2)^2}{2!} - \frac{(x^2/2)^3}{3!} + \dots + (-1)^i \cdot \frac{(x^2/2)^i}{i!} + \dots$	2	2,5	0,1

4.4. ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЯ

Разработать алгоритм и программу вычисления точного значения функции по формуле $y = \sin(x)$ и ее приближенного значения S , представленного бесконечным рядом

$$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{i=0}^{\infty} \frac{(-1)^i \cdot x^{2i+1}}{(2i+1)!}$$

при изменении аргумента функции от $x_{\text{нач}}=3,2$ до $x_{\text{кон}}=4,8$ с шагом $\Delta x=0,2$.

Суммирование членов ряда производить до достижения требуемой точности $\varepsilon = 0,0001$. То есть прибавлять очередной член ряда до тех пор, пока его значение по абсолютной величине не станет меньше ε . Для предотвращения закливания предусмотреть завершение процесса суммирования членов ряда по заданному максимальному номеру члена ряда $n=1000$.

Решение

Алгоритм решения предполагает использование двух вложенных друг в друга циклов. Внешний цикл необходим для перебора последовательности значений аргумента x . Для этого цикла выберем структуру с предусловием: начиная с $x = x_{\text{нач}}$ проводим повторяющиеся действия (вычисляем точное и приближенные значения функции) с последовательным увеличением x на величину шага Δx , пока текущее значение x не превысит $x_{\text{кон}}$. Внутренний цикл необходим для вычисления приближенного значения функции, как суммы сходящегося ряда $S = b + a_0 + a_1 + \dots + a_i + \dots$, где i – порядковый номер члена, b – постоянный начальный член ряда. Для внутреннего цикла выберем структуру цикла с параметром (счетчиком повторений). В качестве параметра возьмем порядковый номер члена i . Номер i будет увеличиваться на единицу после каждого вычисления слагаемого a_i и добавления его в сумму S . Изменение номера ограничим максимальным значением n . Для достижения требуемой точности вычисления S воспользуемся тем свойством сходящегося ряда, что каждый последующий член ряда убывает по абсолютной величине. Выход из цикла организуем с помощью проверки, насколько мала величина очередного члена ряда. Если условие $|a_i| \leq \varepsilon$ выполнится, значит этот текущий член и все следующие за ним добавлять в сумму ряда не нужно и поэтому принудительно выходим из внутреннего цикла. Если указанное условие по любым причинам не выполнится, то внутренний цикл завершится по достижении параметром i максимального значения n . Каждый очередной член ряда будем вычислять по рекуррентной формуле $a_i = R \cdot a_{i-1}$, $i = 1, 2, \dots$, где R – рекуррентный множитель. Выведем формулу для множителя R :

$$a_i = \frac{(-1)^i \cdot x^{2i+1}}{(2i+1)!}; \quad a_{i-1} = \frac{(-1)^{i-1} \cdot x^{2i-1}}{(2i-1)!};$$

$$R = \frac{a_i}{a_{i-1}} = \frac{(-1)^i \cdot x^{2i+1}}{(2i+1)!} \cdot \frac{(2i-1)!}{(-1)^{i-1} \cdot x^{2i-1}} = \frac{-x^2}{(2i+1) \cdot 2i}.$$

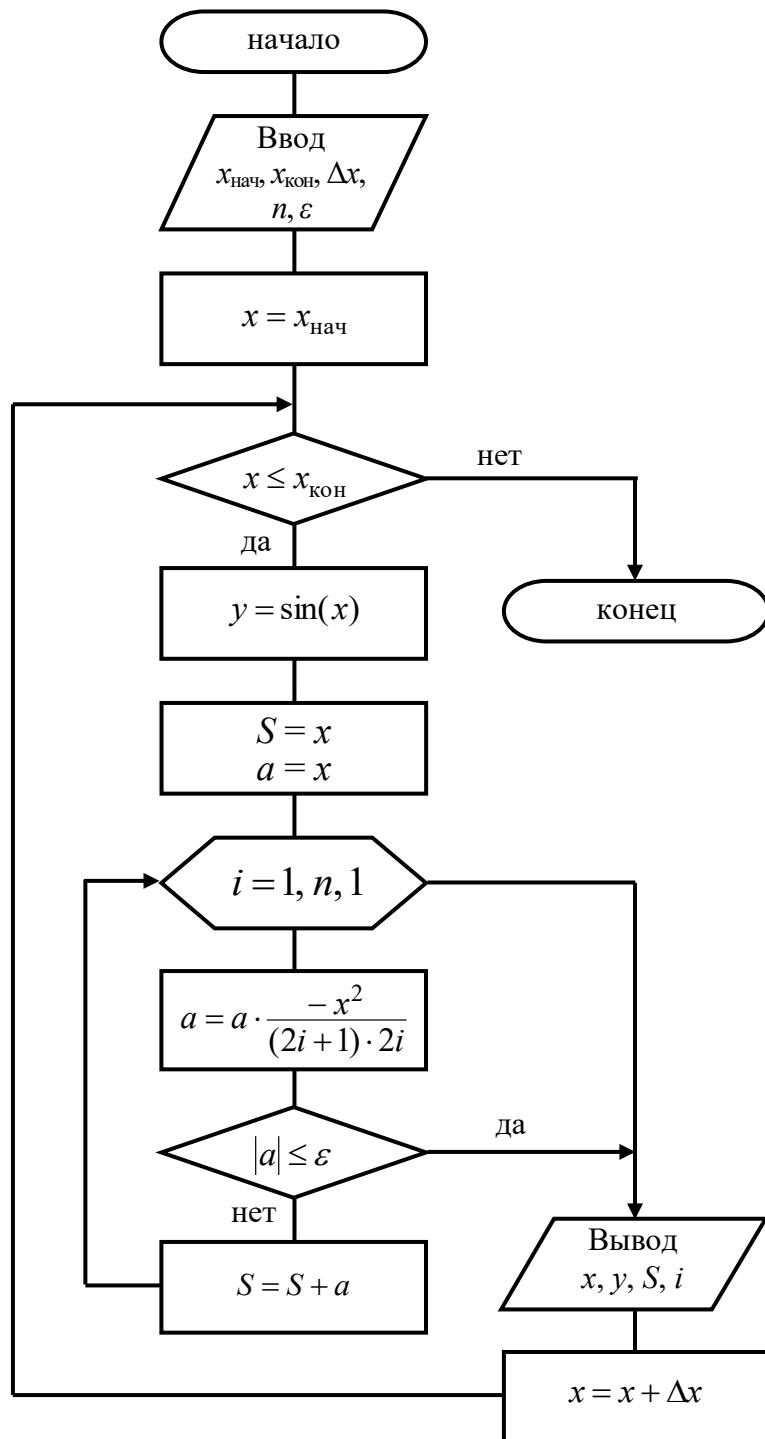
Таким образом имеем рекуррентную формулу

$$a_i = \frac{-x^2}{(2i+1) \cdot 2i} \cdot a_{i-1}, \quad i = 1, 2, \dots$$

Для старта вычислений по этой формуле до начала внутреннего цикла требуется задать значение $a_0 = x$. Соответственно, формулу суммирования можно записать следующим образом

$$S = b + a_0 + \sum_{i=1}^n a_i.$$

До внутреннего цикла также необходимо задать начальное значение переменной S . Это значение $S = b + a_0$. Постоянный начальный член ряда b в задании данного примера отсутствует, поэтому значение переменной S до начала суммирования включает только a_0 , т.е. $S = x$.

Блок-схема алгоритма

Паскаль-программа

```

Program example4;
var x,y,s,xn,xk,dx,a,eps:real;
i,n:integer;
begin
  write ('Введите xn, xk, dx='); readln (xn, xk, dx);
  write ('Введите n='); readln (n);
  write ('Введите eps='); readln (eps);
  x:=xn;
  while x<=xk do
    begin
      y:=sin(x);
      s:=x;
      a:=x;
      for i:=1 to n do
        begin
          a:=a*(-x*x)/((2*i+1)*2*i);
          if abs(a)<eps then break;
          s:=s+a;
        end;
      writeln ('x=',x:1:2,' y=', y:1:5,' s=',s:1:5,' i=',i);
      x:=x+dx;
    end;
end.

```

Выполним рабочий расчет с заданными значениями исходных данных:

```

Окно вывода
Введите xn, xk, dx=3.2 4.8 0.2
Введите n=1000
Введите eps=0.0001
x=3.20 y=-0.05837 s=-0.05835 i=7
x=3.40 y=-0.25554 s=-0.25547 i=7
x=3.60 y=-0.44252 s=-0.44253 i=8
x=3.80 y=-0.61186 s=-0.61188 i=8
x=4.00 y=-0.75680 s=-0.75685 i=8
x=4.20 y=-0.87158 s=-0.87157 i=9
x=4.40 y=-0.95160 s=-0.95159 i=9
x=4.60 y=-0.99369 s=-0.99366 i=9

```

Из результатов видно, что значения y и S совпадают с точностью до требуемой погрешности, следовательно, алгоритм работает правильно.

4.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлении студентом аккуратно оформленного отчета и при умении разрабатывать алгоритмы и программы суммирования функциональных рядов с помощью итерационных циклов.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;
- вывод рекуррентной формулы вычисления члена ряда;
- блок-схему разработанного алгоритма решения задачи;
- текст разработанной программы;
- результаты тестового и рабочего расчета по программе;
- выводы.

4.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какая структура называется вложенными циклами?
2. Что называется «внутренним циклом» и «внешним циклом»?
3. Нарисуйте блок-схему алгоритма, в которой в цикл с предусловием вложен цикл с параметром.
4. За счет чего достигается требуемая точность вычисления суммы сходящегося функционального ряда?
5. Как записывается рекуррентная формула для вычисления текущего члена ряда?
6. Что обеспечивает защиту от «зацикливания» при вычислении суммы ряда в случае, если условие достижения точности не выполняется?

5. РАБОТА №5. ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ С ОДНОМЕРНЫМИ И ДВУМЕРНЫМИ МАССИВАМИ

5.1. ЦЕЛЬ РАБОТЫ

Формирование у студентов умения анализировать исходные данные задачи, овладение практическими навыками разработки алгоритма и программирования операций по обработке одномерных и двумерных массивов.

5.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

Массивом называют упорядоченную последовательность данных одного типа, объединенных под одним именем. Каждое из значений, составляющих массив, называется элементом массива и задается своим порядковым номером в

массиве – так называемым индексом. Размерность массива - количество индексов элементов массива. По размерности массивы делятся на одномерные (векторы), двумерные (матрицы) и т.д. При обращении к вектору индекс указывает на порядковый номер элемента вектора, а в матрице первый индекс указывает на номер строки, второй - на номер столбца, на пересечении которых находится соответствующий элемент. Описание массива в Паскале выглядит так:

Var <Переменная-массив> : **Array** [<Диапазон индексов>] **Of** <Тип элементов>;

Здесь *Диапазон индексов* определяет размерность объявляемого массива.

5.3. ЗАДАНИЯ К РАБОТЕ

Разработать алгоритм и программу решения четырех взаимосвязанных задач по обработке массивов:

- 1) вычисления элементов квадратной матрицы a_{ij} ($i, j = 1, 2, \dots, n$) по заданной формуле;
- 2) вычисления элементов вектора x_i ($i = 1, 2, \dots, n$) по заданному правилу;
- 3) типовой задачи упорядочения элементов матрицы **A** или вектора **X**;
- 4) вычисления значения функции y заданной формуле через элементы матрицы **A** или вектора **X**.

Размерность матрицы и вектора n взять от 4 до 6.

Варианты заданий

Вариант 1	Вариант 2
1) $a_{i,j} = 2,3 \cdot i! - 4,7 \cdot j!$; 2) $x_i = \sum_{j=1}^n a_{ij} \cdot \max_{k=1,n} a_{ik} $; 3) упорядочить элементы побочной диагонали матрицы A по возрастанию значений; 4) $y = \prod_{i=1}^{n-1} \left(\frac{1}{ x_i +1} + x_{i+1} \right)$.	1) $a_{j,i} = \frac{\ln(j+i)}{\ln(i+1)+\ln(j+1)}$; 2) x_i – скалярное произведение i -й строки на i -й столбец; 3) упорядочить нечетные по номеру элементы вектора X по возрастанию значений; 4) $y = \max_{i=1,n} (x_i) - \min_{i=1,n} x_i $.
Вариант 3	Вариант 4
1) $a_{i,j} = \frac{\sqrt[i]{j} + \sqrt[j]{i}}{\sqrt{i \cdot j}}$; 2) x_i – среднееарифметическое значение положительных элементов i -й строки; 3) упорядочить элементы побочной диагонали матрицы A по убыванию абсолютных значений; 4) $y = \max_{i=1,n} (x_i / a_{ii})$.	1) $a_{j,i} = (5 \cdot i + j) \cdot \lg(i + j)$; 2) x_i – среднееарифметическое значение отрицательных элементов i -го столбца; 3) упорядочить элементы первых трех столбцов матрицы A по возрастанию значений; 4) $y = \min_{i=1,n} (x_i / a_{i,n-i+1})$.

Вариант 5 1) $a_{i,j} = (i-5) \cdot j + (j-2)!;$ 2) x_i – скалярное произведение главной диагонали на i-ю строку; 3) упорядочить элементы второй строки матрицы A по возрастанию значений; 4) $y = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i.$ $x_i > 0 \quad x_i < 0$	Вариант 6 1) $a_{j,i} = \frac{\sin(5 \cdot i) + (\cos(10 \cdot i))}{\operatorname{tg}(i \cdot j)};$ 2) x_i – скалярное произведение побочной диагонали на i-й столбец; 3) упорядочить элементы последних трех столбцов матрицы A по возрастанию абсолютных значений; 4) $y = \sum_{i=1}^n \frac{x_i + x_{n-i+1}}{x_{n/2}}.$
Вариант 7 1) $a_{i,j} = e^{-i} \cdot \ln(i \cdot j) + e^{-j};$ 2) $x_i = \min_{j=\overline{1,n}} \left a_{ij} + \sum_{k=1}^n \sin(a_{kj}) \right ;$ 3) упорядочить элементы второго столбца матрицы A по возрастанию значений; 4) $y = \prod_{i=1}^n \left(\frac{x_{n+1-i}}{i} - \sum_{k=1}^n \frac{x_k}{i!} \right).$	Вариант 8 1) $a_{j,i} = \frac{100 \cdot \sin(i) + j}{i^j / j^i};$ 2) x_i – скалярное произведение i-й строки на $(n+1-i)$ столбец; 3) упорядочить элементы второй половины вектора X по убыванию значений; 4) $y = \sum_{i=1}^n x_i - \sum_{i=1}^n (x_i + 1).$ $x_i > 0 \quad x_i < 0$
Вариант 9 1) $a_{i,j} = \frac{(5,2+i) \cdot j}{2 \cdot i + 3 \cdot j};$ 2) $x_i = \prod_{j=1}^n \frac{a_{ij}}{a_{ii}}$ при условии $a_{ii} \neq 0;$ 3) упорядочить элементы последней строки матрицы A по возрастанию значений; 4) $y = \sum_{i=1}^n \sum_{k=1}^i x_k.$	Вариант 10 1) $a_{j,i} = \frac{\ln(i \cdot j) + 2,3}{\ln(i) + \ln(j+1)};$ 2) $x_i = \min_{j=\overline{1,n}} (a_{ij}) + \max_{j=\overline{1,n}} (a_{ji});$ 3) упорядочить четные по номеру элементы вектора X по возрастанию значений; 4) $y = \sum_{i=1}^n x_i \cdot \prod_{k=1}^i x_k$ при условии $x_k \neq 0.$
Вариант 11 1) $a_{i,j} = \frac{2^i + e^j}{3^j};$ 2) $x_i = \sum_{k=1}^i a_{ik};$ 3) упорядочить элементы второй строки матрицы A по возрастанию значений; 4) $y = \sum_{i=1}^n x_i$ при условии $x_i > (x_i + x_n)/2.$	Вариант 12 1) $a_{j,i} = \frac{0,8 \cdot i + \ln(j)}{\sqrt{j!}};$ 2) $x_i = \sum_{j=1,3,\dots}^n \sin(a_{ij}) + \sum_{k=2,4,\dots}^n \cos(a_{ik});$ 3) упорядочить элементы первой половины вектора X по убыванию значений; 4) $y = \prod_{i=1}^n (x_i \cdot \ln x_i ^i).$

Вариант 13 1) $a_{i,j} = 2,7 - i \cdot 2^{-j}$; 2) x_i – скалярное произведение побочной диагонали на i-ю строку; 3) упорядочить элементы четных строк матрицы A по возрастанию значений; 4) $y = \prod_{i=1}^n \frac{i!}{e^i}$.	Вариант 14 1) $a_{j,i} = \frac{-2^i}{n^2 + 1 - i \cdot j}$; 2) x_i – скалярное произведение k-го столбца на i-ю строку, где k- номер максимального элемента i-й строки; 3) упорядочить элементы столбцов матрицы A по возрастанию абсолютных значений; 4) $y = \sum_{i=1}^n x_i \cdot x_{n+1-i}$.
Вариант 15 1) $a_{i,j} = \sqrt[i]{(i+1) \cdot j} \cdot \sqrt[j]{(j+1) \cdot i}$; 2) $x_i = \max_{j=\overline{1,n}} (a_{ji}^j \cdot a_{ij}^i)$; 3) упорядочить элементы каждой строки матрицы A по возрастанию значений; 4) $y = \prod_{i=1}^{n-1} (x_i + x_{i+1})$.	Вариант 16 1) $a_{j,i} = (-1)^{i \cdot j} \cdot \ln(i \cdot j + 1)$; 2) x_i – скалярное произведение i-ГО столбца на последнюю строку; 3) упорядочить элементы первых трех строк матрицы A по убыванию значений; 4) $y = \sum_{i=1}^n x_{n+1-i} \cdot \sum_{k=1}^i x_k$.
Вариант 17 1) $a_{i,j} = \frac{2 \cdot i! + 4}{(n - j + 1) \cdot 3 \cdot j + 5}$; 2) $x_i = \prod_{j=1}^n (\operatorname{tg}(a_{ij}) + \sum_{k=j}^n a_{kj})$; 3) упорядочить элементы предпоследней строки матрицы A по возрастанию абсолютных значений; 4) $y = \prod_{i=1}^n x_i$ при условии $x_i > 0$.	Вариант 18 1) $a_{j,i} = i - j ^{-3} \cdot \ln(i)$; 2) x_i – скалярное произведение i-го столбца на k-ю строку, где k-номер максимального элемента второй строки; 3) упорядочить элементы четных столбцов матрицы A по возрастанию значений; 4) $y = \prod_{i=1}^n \frac{x_i}{i+1}$ при условии $x_i \neq 0$.
Вариант 19 1) $a_{i,j} = \frac{3 \cdot i / i^3 + 3^j / j^3}{e^j / e^i}$; 2) x_i – скалярное произведение i-й строки на i-й столбец; 3) упорядочить элементы второй половины вектора X по убыванию абсолютных значений; 4) $y = \prod_{i=1}^n \frac{x_{n+1-i}}{i!}$.	Вариант 20 1) $a_{j,i} = \frac{(-1)^{i+j} \cdot (4,3 - i)}{ 0,2 - i \cdot j \cdot n}$; 2) x_i – среднеарифметическое значение положительных элементов i-й строки; 3) упорядочить элементы каждого столбца матрицы A по возрастанию значений; 4) $y = (\max_{i=\overline{1,n}}(x_i) - \min_{i=\overline{1,n}}(x_i))/2$.

Вариант 21 1) $a_{i,j} = \frac{10^{\ln(i)} / 10^{\ln(j)}}{e^{\lg(i)} / e^{\lg(j)}};$ 2) $x_i = \max(a_{ij} \cdot \cos(a_{ij}));$ $j = \overline{1, n}$ 3) упорядочить элементы главной диагонали матрицы A по возрастанию абсолютных значений; 4) $y = \prod_{i=1}^n \left(x_i \cdot \sum_{k=i}^n x_{n+1-k} \right).$	Вариант 22 1) $a_{j,i} = \frac{e^{i+j}}{10^{i+j/2}};$ 2) x_i – среднеарифметическое значение элементов i-го столбца; 3) упорядочить элементы последних трех строк матрицы A по убыванию значений; 4) $y = \prod_{i=1}^n \left(\sum_{k=1}^i x_k \right) \cdot \left(\sum_{k=i}^n x_k \right).$
Вариант 23 1) $a_{i,j} = (2 \cdot i^2 + j^2/2) + \ln(i \cdot j + 1);$ 2) x_i – скалярное произведение i-й строки на вектор B, где $b_k = \prod_{j=1}^n a_{kj} - \min(a_{kj}); \quad k = \overline{1, n};$ $j = \overline{1, n}$ 3) упорядочить элементы второй половины вектора X по возрастанию значений; $y = \sum_{i=1}^n \left(x_i \cdot \sum_{k=1}^n b_{n+1-k} \right)$	Вариант 24 1) $a_{j,i} = \frac{\ln(i+1) + \ln(j+2)}{\ln(i \cdot j + 3)};$ 2) $x_i = \frac{\sqrt{a_{1i}^2 + a_{2i}^2 + \dots + a_{ni}^2} / n}{n};$ 3) упорядочить элементы первого столбца матрицы A по убыванию значений; 4) $y = \max_{i=\overline{1, n}} (x_i) - \prod_{i=1}^n (x_i - 2)$ $i=\overline{1, n} \quad x_i \neq 2$
Вариант 25 1) $a_{i,j} = \frac{ i-j - j-i }{i^2 - 2 \cdot i \cdot j + j^2 + 1};$ 2) x_i – элементы побочной диагонали матрицы, полученной из матрицы A перестановкой строк в соответствии с возрастанием элементов первого столбца; 3) упорядочить элементы последних трех строк матрицы A по возрастанию абсолютных значений; 4) $y = \prod_{i=1}^n (x_i - \sin(x_i)).$	Вариант 26 1) $a_{j,i} = \frac{\sin(i) + \cos(j)}{\operatorname{tg}(i) + \operatorname{ctg}(j)};$ 2) x_i – элементы главной диагонали матрицы, полученной из матрицы A заменой отрицательных элементов нулями; 3) упорядочить элементы четных столбцов матрицы A по убыванию абсолютных значений; 4) $y = \sum_{i=n}^1 \left(x_i \cdot \sum_{k=1}^i x_k \right).$

Вариант 27 1) $a_{i,j} = \sqrt{i^3 + j^2 + i \cdot j} / \sqrt{j^4 + i^3 + 2 \cdot i \cdot j}$; 2) x_i – элементы побочной диагонали матрицы, полученной из матрицы A перестановкой 2 и 4 строк; 3) упорядочить элементы второго столбца матрицы A по убыванию абсолютных значений; 4) $y = \prod_{i=1}^n \left(x_i \cdot \sum_{k=1}^n x_k \right)$. $x_i \neq 0$	Вариант 28 1) $a_{j,i} = \frac{(n-j) \cdot (n-i)}{ 2 \cdot n - i - j + 1 }$; 2) $x_i = \sum_{k=2}^n a_{ik} + \sum_{l=1}^i a_{li}$, где k – четно, l – нечетно; 3) упорядочить элементы второй половины вектора X по возрастанию значений; 4) y – среднеарифметическое значение ненулевых элементов вектора X.
Вариант 29 1) $a_{i,j} = (i+j) \cdot \sin(i \cdot j) ^3$; 2) $x_i = \sum_{j=1}^n a_{ij}$ для всех $a_{ij} \in [2;10]$; 3) упорядочить элементы главной диагонали матрицы A по убыванию значений; 4) $y = \prod_{i=2}^{n-1} \sum_{k=i}^{n-1} (x_{k-1} + x_{k+1})/2$.	Вариант 30 1) $a_{j,i} = i \cdot \operatorname{tg}(i) + j \cdot \operatorname{tg}(j)$; 2) x_i – скалярное произведение побочной диагонали на i-ю строку; 3) упорядочить элементы строк матрицы A по возрастанию значений; 4) $y = \prod_{i=1}^n x_i ^i \cdot \ln x_{n-i+1}$. $x_i > 0$; $x_{n-i+1} \neq 0$

5.4. ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЯ

Разработать алгоритм и программу решения четырех взаимосвязанных задач:

- 1) вычисление элементов квадратной матрицы A по формуле

$$a_{ij} = \frac{2ij + 4i}{j}; \quad i, j = 1, 2, \dots, n;$$

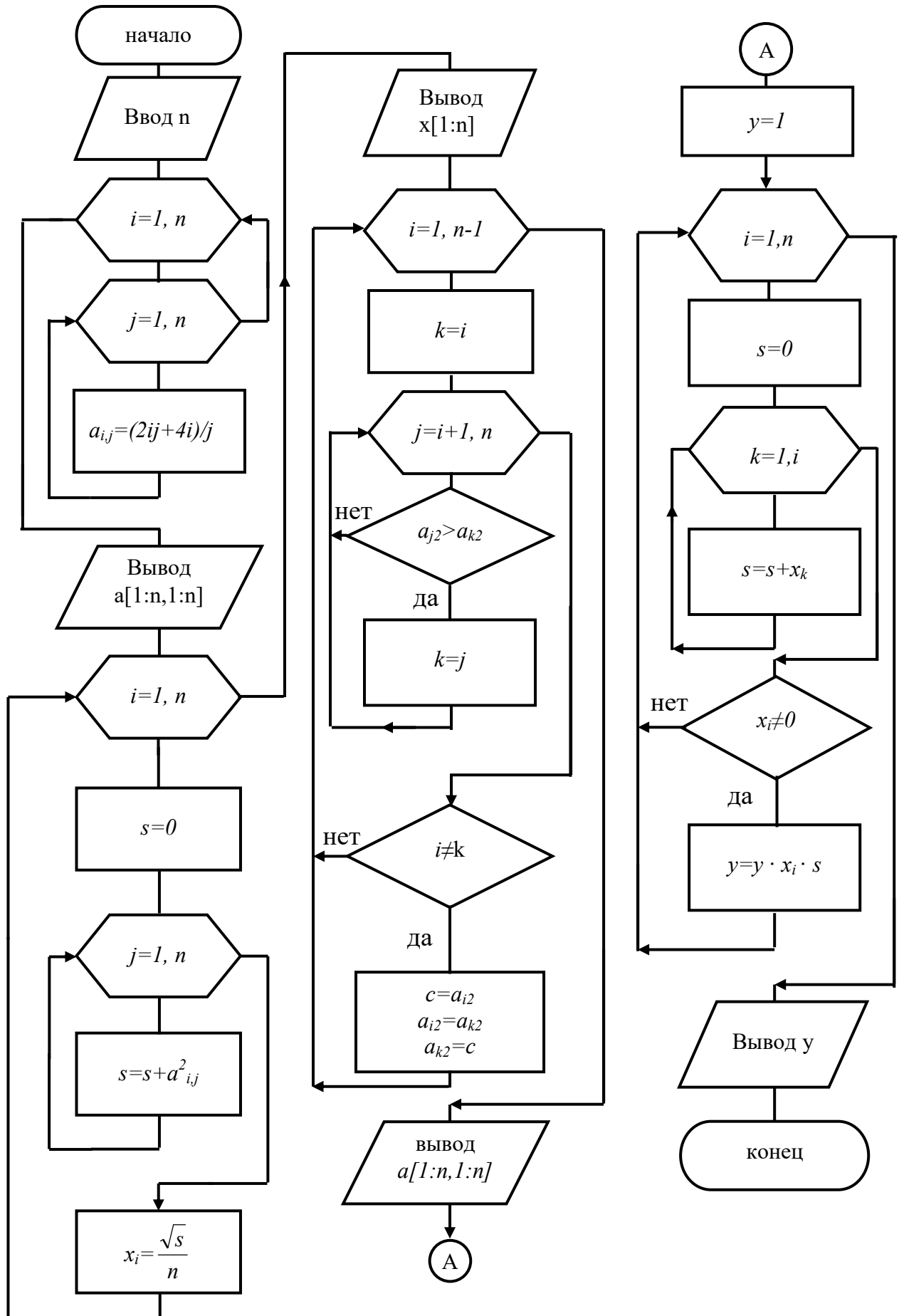
- 2) вычисление элементов вектора X по следующему правилу:

$$x_i = \frac{\sqrt{\sum_{j=1}^n a_{i,j}^2}}{n}; \quad i = 1, 2, \dots, n;$$

- 3) упорядочить элементы второго столбца матрицы A по убыванию значений;

- 4) вычислить значение функции y по формуле $y = \prod_{i=1}^n x_i \sum_{k=1}^i x_k$ при условии $x_i \neq 0$.

Предварительно провести отладку программы при $n = 3$ (вычисления провести вручную или с помощью калькулятора), затем проверить полученные данные на компьютере.

Блок-схема алгоритма

Паскаль-программа

```

program example5;
  var a:array[1..10,1..10] of real;
  x:array [1..10] of real;
  i,j,k,n:integer;
  y,s,c:real;
begin
  writeln('Введите n'); read(n);
  for i:=1 to n do                                {начало 1 задачи}
    for j:=1 to n do
      a[i,j]:=(2*j*i+4*i)/j;
  writeln('  :20,'МАТРИЦА A');
  for i:=1 to n do
    begin
      for j:=1 to n do write(a[i,j]:8:3);
      writeln;
    end;
  for i:=1 to n do                                {начало 2 задачи}
    begin
      s:=0;
      for j:=1 to n do s:=s+sqr(a[i,j]);
      x[i]:=sqrt(s)/n;
    end;
  writeln('  :20,'ВЕКТОР X');
  for i:=1 to n do write(x[i]:8:3);
  writeln;
  for i:=1 to n-1 do                                {начало 3 задачи}
    begin
      k:=i;
      for j:=i+1 to n do
        if a[j,2]>a[k,2] then k:=j;
      if i<>k then
        begin
          c:=a[i,2]; a[i,2]:=a[k,2]; a[k,2]:=c;
        end;
      end;
  writeln('  :14,'УПОРЯДОЧЕННАЯ МАТРИЦА');
  for i:=1 to n do
    begin
      for j:=1 to n do write(a[i,j]:8:3);
      writeln;
    end;
  writeln;
  y:=1;                                            {начало 4 задачи}
  for i:=1 to n do
    begin

```

```

s:=0;
for k:=1 to n do s:=s+x[k];
if x[i]<>0 then y:=y*x[i]*s;
end;
writeln(' ЗНАЧЕНИЕ ФУНКЦИИ Y=',y:1:3);
end.

```

Выполним рабочий расчет для размерности матрицы $n = 6$:

Окно вывода						
Введите n						
6						
МАТРИЦА А						
6.000	4.000	3.333	3.000	2.800	2.667	
12.000	8.000	6.667	6.000	5.600	5.333	
18.000	12.000	10.000	9.000	8.400	8.000	
24.000	16.000	13.333	12.000	11.200	10.667	
30.000	20.000	16.667	15.000	14.000	13.333	
36.000	24.000	20.000	18.000	16.800	16.000	
ВЕКТОР X						
1.555	3.110	4.665	6.220	7.776	9.331	
УПОРЯДОЧЕННАЯ МАТРИЦА						
6.000	24.000	3.333	3.000	2.800	2.667	
12.000	20.000	6.667	6.000	5.600	5.333	
18.000	16.000	10.000	9.000	8.400	8.000	
24.000	12.000	13.333	12.000	11.200	10.667	
30.000	8.000	16.667	15.000	14.000	13.333	
36.000	4.000	20.000	18.000	16.800	16.000	
ЗНАЧЕНИЕ ФУНКЦИИ Y=12354118108332.900						

Из результатов видно, что алгоритм работает правильно.

5.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлении студентом аккуратно оформленного отчета и при умении разрабатывать алгоритмы и программы для обработки одномерных и двумерных массивов.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;
- блок-схему разработанного алгоритма решения задачи;
- текст разработанной программы;
- результаты ручного и компьютерного расчетов;
- выводы.

5.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое «массив», «элемент массива», «индекс элемента массива»?
2. Что такое «размерность» массива? Назовите виды массивов.
3. Как объявить одномерный целочисленный массив А из 100 элементов?
4. Как объявить двумерный вещественный массив А из 50 элементов?
5. Какие операторы Паскаля можно использовать для перебора элементов одномерного и двумерного массивов?
6. Опишите словесно алгоритм сортировки элементов одномерного массива.
7. Запишите алгоритм вычисления скалярного произведения двух векторов.

6. РАБОТА №6. ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНОГО УРАВНЕНИЯ

6.1. ЦЕЛЬ РАБОТЫ

Целью работы является приобретение студентами умения приближенного решения на компьютере алгебраических и трансцендентных уравнений.

6.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

6.2.1. Общие сведения

В инженерной практике часто встречается задача по отысканию корней уравнения

$$f(x) = 0, \quad (1)$$

где $f(x)$ – произвольная функция. Выбор подходящего метода решения таких уравнений зависит от характера рассматриваемой задачи.

Нелинейные уравнения можно разделить на два класса – алгебраические и трансцендентные. *Алгебраическими уравнениями* называют уравнения, содержащие только алгебраические функции (целые, рациональные, иррациональные). В частности, многочлен является целой алгебраической функцией. Уравнения, содержащие другие функции (тригонометрические, показательные, логарифмические и другие), называются *трансцендентными*.

Для численного решения нелинейных уравнений применяют только итерационные методы.

Задача нахождения корня уравнения $f(x) = 0$ итерационным методом состоит из двух этапов:

- *отделение корней* - отыскание приближенного значения корня или содержащего его отрезка;
- *уточнение приближенных корней* - доведение их до заданной степени точности.

Процесс отделения корней начинается с установления знаков функции $f(x)$ в граничных $x = a$ и $x = b$ точках области ее существования.

Пример 1. Отделить корни уравнения:

$$f(x) \equiv x^3 - 6x + 2 = 0. \quad (2)$$

Составим приблизительную схему:

x	$-\infty$	-3	-1	0	1	3	$+\infty$
$f(x)$	$-$	$-$	$+$	$+$	$-$	$+$	$+$

Следовательно, уравнение (2) имеет три действительных корня, лежащих в интервалах $[-3, -1]$, $[0, 1]$ и $[1, 3]$.

Приближенные значения корней (*начальные приближения*) могут быть также известны из физического смысла задачи, из решения аналогичной задачи при других исходных данных.

В инженерной практике распространен *графический способ* определения приближенных корней.

Принимая во внимание, что действительные корни уравнения (1) - это точки пересечения графика функции $f(x)$ с осью абсцисс, достаточно построить график функции $f(x)$ и отметить точки пересечения $f(x)$ с осью Ox или отметить на оси Ox отрезки, содержащие по одному корню. Построение графиков часто удается сильно упростить, заменив уравнение (1) *равносильным* ему уравнением

$$f_1(x) = f_2(x), \quad (3)$$

где функции $f_1(x)$ и $f_2(x)$ - более простые, чем функция $f(x)$. Тогда, построив графики функций $y = f_1(x)$ и $y = f_2(x)$, искомые корни получим как абсциссы точек пересечения этих графиков.

Пример 2. Графически отделить корни уравнения (рис. 4)

$$x \lg x = 1. \quad (4)$$

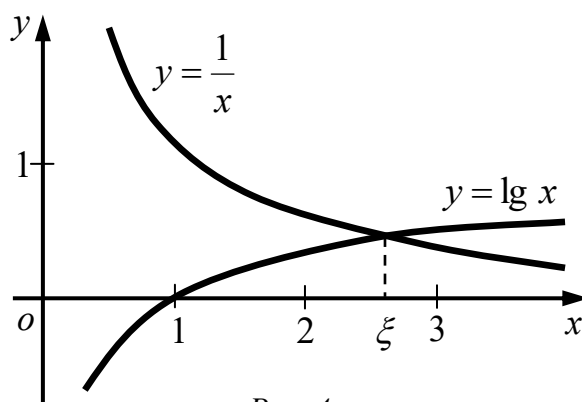


Рис. 4

Уравнение (4) удобно переписать в другом виде:

$$\lg x = \frac{1}{x}.$$

Отсюда ясно, что корни уравнения (4) могут быть найдены как абсциссы точек пересечения логарифмической кривой $y = \lg x$ и гиперболы $y = \frac{1}{x}$.

Построив эти кривые, приближенно найдем единственный корень $\xi \approx 2,5$ уравнения (4) или определим содержащий его отрезок $[2, 3]$.

Далее рассмотрим вычислительные методы для уточнения приближенных значений корней уравнения $f(x) = 0$.

6.2.2. Метод половинного деления*

Для нахождения корня уравнения (1), принадлежащего отрезку $[a, b]$, делим этот отрезок пополам: $x = (a + b)/2$. Если $f(x) = 0$, то $\xi = x$ является корнем уравнения, но такая ситуация маловероятна. Если $f(x) \neq 0$, то выбираем ту из половин $[a, x]$ или $[x, b]$, на концах которой функция $f(x)$ имеет противоположные знаки и, следовательно, содержит корень уравнения. Для проверки знаков воспользуемся условием $f(a) \cdot f(x) < 0$. Половину отрезка, не содержащую корень, отбрасываем путем переноса соответствующей границы отрезка в точку x . Новый суженный отрезок $[a^1, b^1]$ снова делим пополам и производим те же самые действия. Итерационный процесс продолжается до тех пор, пока не будет выполнено условие

$$|b^i - a^i| < \varepsilon,$$

где ε – заданная предельная абсолютная погрешность, i – номер итерации.

Схема алгоритма уточнения корня методом половинного деления приведена на рис. 5.

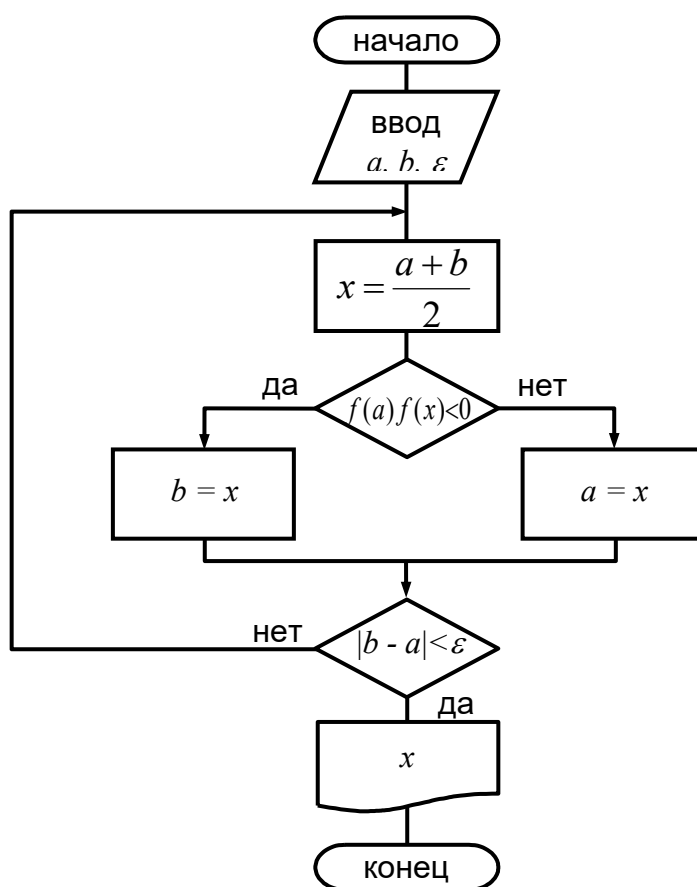


Рис. 5. Алгоритм метода половинного деления

* В литературе встречается другое название – метод дихотомии.

Метод половинного деления прост и надежен, всегда сходится. Недостатком этого метода является низкая скорость сходимости.

Пример 3. Методом половинного деления уточнить корень уравнения

$$f(x) \equiv x^4 + 2x^3 - x - 1 = 0,$$

лежащий на отрезке $[0, 1]$.

Последовательно имеем:

$$f(0) = -1; f(1) = 1;$$

$$f(0,5) = 0,06 + 0,25 - 0,5 - 1 = -1,19;$$

$$f(0,75) = 0,32 + 0,84 - 0,75 - 1 = -0,59;$$

$$f(0,875) = 0,59 + 1,34 - 0,88 - 1 = +0,05;$$

$$f(0,8125) = 0,436 + 1,072 - 0,812 - 1 = -0,304;$$

$$f(0,8438) = 0,507 + 1,202 - 0,844 - 1 = -0,135;$$

$$f(0,8594) = 0,546 + 1,270 - 0,859 - 1 = -0,043 \text{ и т. д.}$$

Можно принять

$$\xi = \frac{1}{2}(0,859 + 0,875) = 0,867.$$

6.2.3. Комбинированный метод (метод хорд и касательных)

Идея метода заключается в том, что кривая $f(x)$ на интервале поиска заменяется прямыми линиями (хордой и касательной) и в качестве приближений к истинному корню уравнения (1) берутся точки пересечения этих прямых с осью абсцисс. Геометрически метод сводится к тому, что приближение к истинному значению корня на каждой итерации происходит одновременно с двух сторон интервала $[a, b]$. При этом для приближения к корню с одной стороны строится хорда, а с другой – касательная.

Пусть для определенности $f'(x) > 0$ и $f''(x) > 0$ при $a \leq x \leq b$. Тогда для приближения к корню со стороны границы a используем построение хорды, а со стороны b – касательной (рис. 6).

На первой итерации строим хорду A_0B_0 и проводим касательную в точке B_0 . Левую границу интервала a переносим в точку пересечения хорды и оси Ox . Из уравнения хорды имеем

$$a_1 = a - \frac{f(a)}{f(b) - f(a)}(b - a).$$

Соответственно правую границу переносим в точку пересечения касательной с осью Ox и, используя уравнение касательной, получим

$$b_1 = b - \frac{f(b)}{f'(b)}.$$

Таким образом, используя формулы хорд и касательных, на первой итерации мы сузили интервал изоляции корня до отрезка $[a_1, b_1]$.

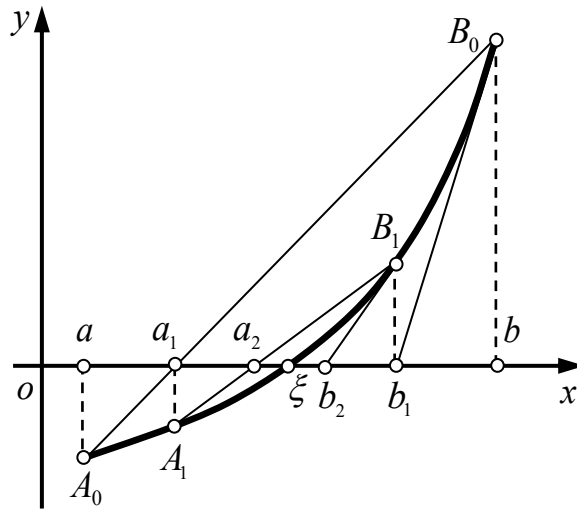


Рис. 6. Комбинированный метод

На следующей итерации проводим хорду A_1B_1 , касательную в точке B_1 и сужаем интервал до отрезка $[a_2, b_2]$.

Продолжая этот процесс, для $i+1$ итерации получим формулы:

$$a_{i+1} = a_i - \frac{f(a_i)}{f(b_i) - f(a_i)}(b_i - a_i), \quad (5)$$

$$b_{i+1} = b_i - \frac{f(b_i)}{f'(b_i)}. \quad (6)$$

Сужение интервала изоляции корня следует проводить до тех пор, пока он не станет меньше заданной предельной погрешности ε :

$$|b_{i+1} - a_{i+1}| < \varepsilon.$$

За значение корня можно принять среднее арифметическое полученных границ интервала.

Если данный метод используется для случая, когда на интервале $[a, b]$ $f'(x) > 0$ и $f''(x) < 0$, то формулы (5) и (6) примут следующий вид:

$$b_{i+1} = b_i - \frac{f(b_i)}{f(b_i) - f(a_i)}(b_i - a_i),$$

$$a_{i+1} = a_i - \frac{f(a_i)}{f'(a_i)}.$$

Схема алгоритма уточнения корня комбинированным методом приведена на рис. 7.

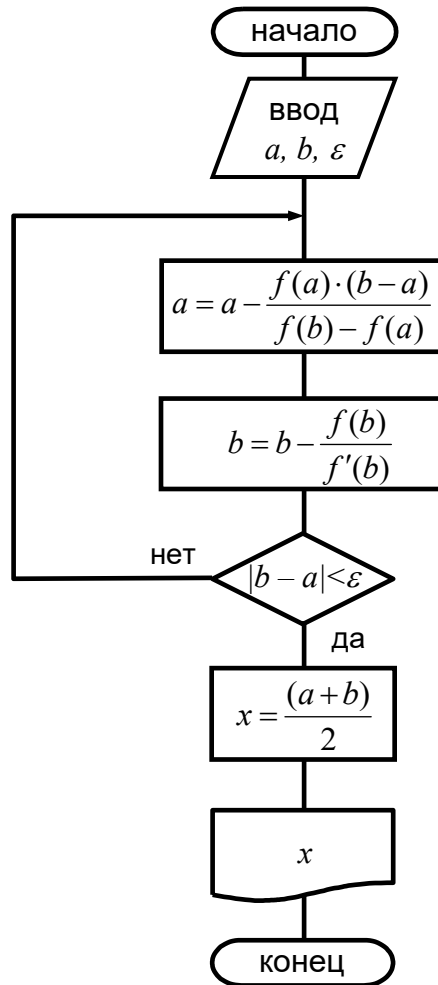


Рис. 7. Алгоритм комбинированного метода

6.2.4. Метод простой итерации

Для использования метода итерации исходное нелинейное уравнение $f(x) = 0$ заменяется равносильным уравнением

$$x = g(x). \quad (7)$$

Пусть известно начальное приближение корня $x = x_0$. Подставляя это значение в правую часть уравнения (7), получим новое приближение:

$$x_1 = g(x_0).$$

Далее, подставляя каждый раз новое значение корня в (7), получаем последовательность значений:

$$x_{i+1} = g(x_i), \quad (i = 0, 1, \dots). \quad (8)$$

Геометрически метод итерации может быть пояснен следующим образом. Построим на плоскости xOy графики функций $y = x$ и $y = g(x)$. Каждый дейст-

вительный корень ξ уравнения (7) является абсциссой точки пересечения M кривой $y = g(x)$ с прямой $y = x$ (рис. 8,а).

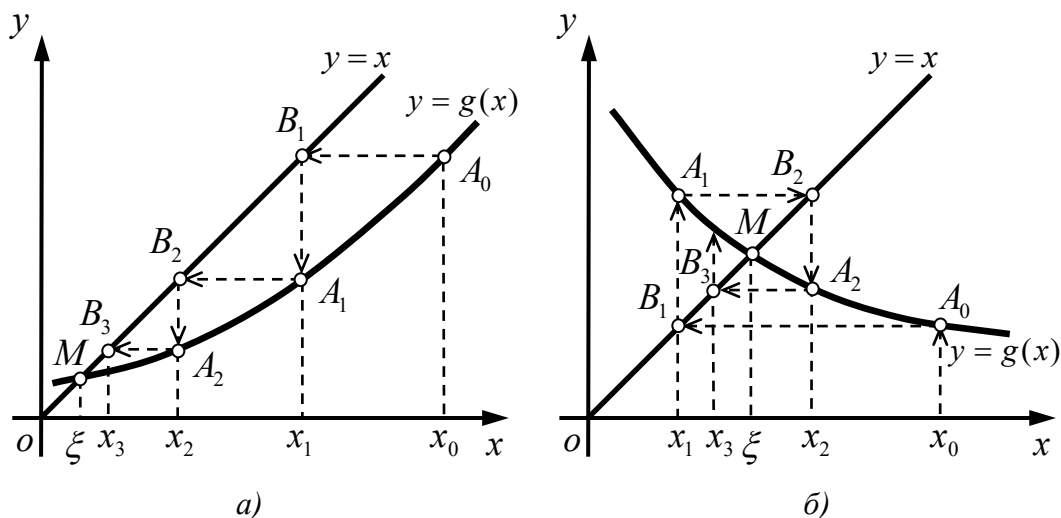


Рис. 8. Сходящиеся итерационные процессы

Отправляясь от некоторой точки $A_0 [x_0, g(x_0)]$, строим ломаную $A_0B_1A_1B_2A_2\dots$ («лестница»), звенья которой попеременно параллельны оси Ox и оси Oy , вершины $A_0, A_1, A_2, \dots$ лежат на кривой $y=g(x)$, а вершины $B_1, B_2, B_3, \dots$, - на прямой $y = x$. Общие абсциссы точек A_1 и B_1, A_2 и $B_2, \dots$, очевидно, представляют собой соответственно последовательные приближения $x_1, x_2, \dots$ корня ξ . Возможен также другой вид ломаной $A_0B_1A_1B_2A_2\dots$ – «спираль» (рис. 8,б). Решение в виде «лестницы» получается, если производная $g'(x)$ положительна, а решение в виде «спирали», если $g'(x)$ отрицательна.

На рис. 8, а, б кривая $y = g(x)$ в окрестности корня ξ - пологая, то есть $|g'(x)| < 1$, и процесс итерации сходится. Однако, если рассмотреть случай, где $|g'(x)| > 1$, то процесс итерации будет расходящимся.

Таким образом, в методе итераций важно правильно выбрать вид функции $g(x)$ и значение начального приближения. Схема алгоритма метода простой итерации показана на рис. 9.

Пример 4. Уравнение

$$f(x) \equiv x^3 - x - 1 = 0 \quad (9)$$

имеет корень $\xi \in [1, 2]$, так как $f(1) = -1 < 0$ и $f(2) = 5 > 0$.

Уравнение (13) можно записать в виде

$$x = x^3 - 1. \quad (10)$$

Здесь $g(x) = x^3 - 1$ и $g'(x) = 3x^2$, поэтому $g'(x) \geq 3$ при $1 \leq x \leq 2$ и, следовательно, условия сходимости процесса итерации не выполнены.

Если записать уравнение (9) в виде

$$x = \sqrt[3]{x+1}, \quad (11)$$

то будем иметь:

$$g(x) = \sqrt[3]{x+1} \quad \text{и} \quad g'(x) = \frac{1}{3\sqrt[3]{(x+1)^2}}.$$

Отсюда $0 < g'(x) < \frac{1}{3\sqrt[3]{4}} < \frac{1}{4}$ при $1 \leq x \leq 2$ и, значит, процесс итерации для уравнения (11) быстро сойдется.

Найдем корень ξ уравнения (9) с точностью до 10^{-2} . Вычисляем последовательные приближения x_n с одним запасным знаком по формуле

$$x_i = \sqrt[3]{x_i + 1} \quad (i = 0, 1, 2, \dots).$$

Найденные значения помещены в табл. 1:

Таблица 1

Значения последовательных приближений x_i

i	0	1	2	3	4
x_i	1	1,260	1,312	1,322	1,3243

С точностью до 10^{-2} можно положить $\xi = 1,324$.

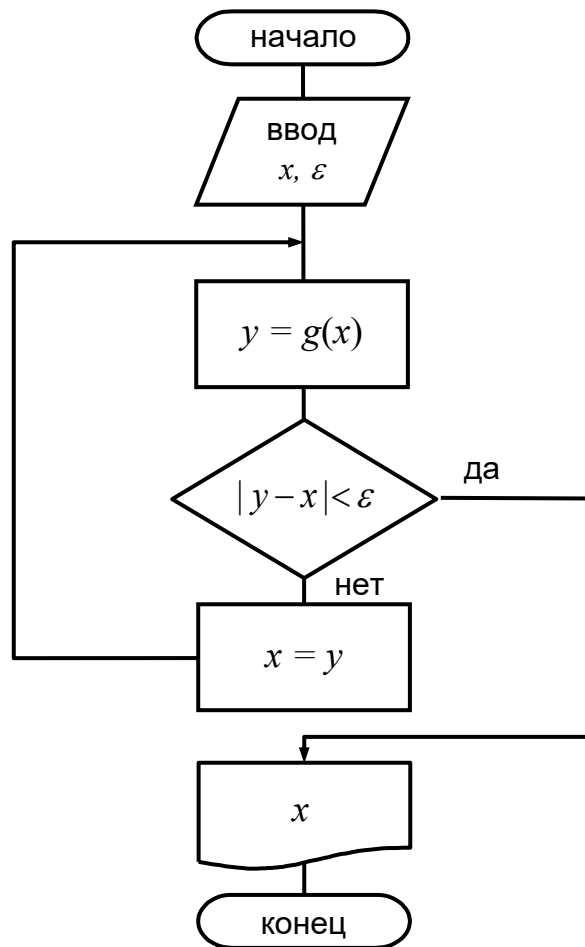


Рис. 9. Алгоритм метода простой итерации

6.3. ЗАДАНИЯ К РАБОТЕ

Дано уравнение $f(x)=0$. Найти с точностью $\varepsilon = 0,0001$ корни этого уравнения (в случае нескольких корней – уточнить значение одного из них). Для уточнения корней использовать методы: половинного деления (дихотомии), хорд и касательных (комбинированный), простой итерации.

Варианты заданий

№ варианта	Уравнение	№ варианта	Уравнение
1	$2^x + 5x - 3 = 0$	2	$\arctg x - \frac{1}{3x^3} = 0$
3	$5^x + 3x = 0$	4	$2e^x + 5x = 0$
5	$3^{x-1} - 2 - x = 0$	6	$2\arctg x - \frac{7}{2x^3} = 0$
7	$e^{-2x} - 2x + 1 = 0$	8	$5^x - 6x - 3 = 0$
9	$\arctg(x-1) + 2x = 0$	10	$\arccctg x - x + 3 = 0$
11	$3^x + 2x - 2 = 0$	12	$\arctg x - 3x + 2 = 0$
13	$2^x + 2x - 5 = 0$	14	$2e^x - 3x - 2 = 0$
15	$3^{x-1} - 4 - x = 0$	16	$4x - 5\ln x - 5 = 0$
17	$e^x + x + 1 = 0$	18	$3^x - 2x - 5 = 0$
19	$\arctg(x-1) + 3x - 1 = 0$	20	$2\arccctg x - x + 3 = 0$
21	$\cos x - 2x = 0$	22	$\arccctg x + 2x - 1 = 0$
23	$3^x + 2x - 3 = 0$	24	$e^x - 5x = 0$
25	$3^x - x - 2 = 0$	26	$\arccctg(x-1) + 2x - 3 = 0$
27	$e^{-2x} - 2x - 2 = 0$	28	$x - e^{-x} = 0$
29	$\arctg(x+1) - 2x = 0$	30	$3^x + 5x - 3 = 0$

6.4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теорией приближенного решения нелинейных уравнений.
2. Отделить корни уравнения $f(x)=0$ сначала аналитически (см. пример 1), а затем графически (см. пример 2). Для каждого корня указать интервал изоляции $[a, b]$.
3. Уточнить значение выбранного корня с точностью ε методом половинного деления (использовать программу 1, приложение 1).
4. Уточнить значение корня с точностью ε методом хорд и касательных (использовать программу 2, приложение 1).
5. Сравнить скорость сходимости (по числу итераций) метода половинного деления с методом хорд и касательных.
6. Уравнение $f(x)=0$ привести к виду $x=\varphi(x)$, удобному для итераций. Проверить достаточное условие сходимости метода (пример 4).
7. Уточнить значение корня с точностью ε методом простой итерации (использовать программу 3, приложение 1).
8. Сравнить число итераций в обоих случаях, а также сравнить скорость сходимости с двумя предыдущими методами. Объяснить результаты.
9. Оформить отчет по лабораторной работе.

6.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлении студентом аккуратно оформленного отчета и при умении применять на практике численные методы решения нелинейных уравнений.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием лабораторной работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;
- графический материал;
- схемы алгоритмов применяемых методов;
- тексты используемых программ;
- результаты вычислительного эксперимента;
- анализ полученных результатов.

6.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Постановка задачи решения нелинейных уравнений. Основные этапы решения задачи.
2. Чем отличаются точные и итерационные методы решения?
3. В чем заключается операция отделения корней уравнения?
4. Какие способы отделения корней существуют?
5. Как проводится графическое отделение корней?

6. Метод половинного деления: описание метода, скорость сходимости, критерий окончания.
7. Метод хорд: описание метода, критерий окончания, геометрическая иллюстрация.
8. Метод Ньютона решения нелинейного уравнения: описание метода, теорема о сходимости, критерий окончания, геометрическая иллюстрация.
9. Комбинированный метод: описание метода, критерий окончания, геометрическая иллюстрация.
10. Метод простой итерации: описание метода, условие сходимости, критерий окончания, геометрическая иллюстрация, приведение к виду, удобному для итераций.
11. Критерий сходимости метода простой итерации. Теорема сходимости.
12. Достоинства и недостатки различных методов уточнения корня.

7. РАБОТА №7. ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМЫ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

7.1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение студентами умения приближенного решения на компьютере системы нелинейных уравнений.

7.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

7.2.1. Общие сведения

В отличие от систем линейных уравнений для систем нелинейных уравнений неизвестны прямые методы решения. Лишь в отдельных случаях систему можно решить непосредственно. Например, для системы из двух уравнений иногда удастся выразить одно неизвестное через другое и таким образом свести задачу к решению одного нелинейного уравнения относительно одного неизвестного. Поэтому итерационные методы для нелинейных систем приобретают особую актуальность.

В наиболее общем случае систему нелинейных уравнений можно представить в виде

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0; \\ f_2(x_1, x_2, \dots, x_n) = 0; \\ \dots \quad \dots \quad \dots \quad \dots \\ f_n(x_1, x_2, \dots, x_n) = 0, \end{cases} \quad (12)$$

т.е. как n функций f_i от n неизвестных x_i .

Требуется найти значения x_i , удовлетворяющие всем уравнениям данной системы.

7.2.2. Метод простой итерации

Метод простой итерации для решения нелинейных уравнений, по существу, является развитием метода простой итерации для одного уравнения. Он основан на допущении, что систему уравнений можно привести к виду

$$\begin{cases} x_1^* = g_1(x_1, x_2, \dots, x_n); \\ x_2^* = g_2(x_1^*, x_2, \dots, x_n); \\ \dots \dots \dots \dots \dots \\ x_n^* = g_n(x_1^*, x_2^*, \dots, x_{n-1}^*, x_n). \end{cases} \quad (13)$$

Алгоритм метода представлен на рис. 10 и заключается в следующем. Используя исходные приближения x_i , из уравнений (13) последовательно

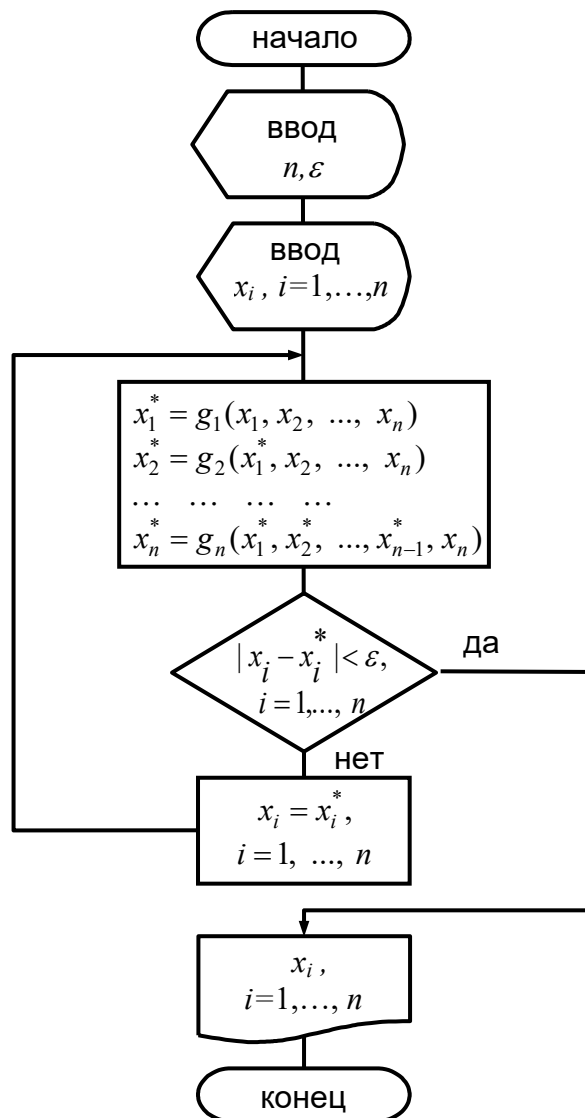


Рис. 10. Алгоритм метода простых итераций для системы уравнений

вычисляют новые значения x_i^* , при этом в правой части каждого последующего j -го уравнения все переменные x_i ($i < j$) заменяют на новые значения x_i^* , вычисленные по предыдущим уравнениям. Значения x_i^* сравнивают с предыдущими значениями x_i и выясняют, достаточно ли мало различие между ними. Если каждая переменная изменилась лишь на достаточно малую величину, то расчет прекращают. Если же это изменение больше заданной величины погрешности ε , процесс повторяют, причем значения x_i^* используют в качестве новых исходных значений неизвестных.

Хотя этот метод прямо ведет к решению, у него есть недостаток. Так, если исходные значения переменных слишком сильно отличаются от истинного решения, то процесс не сойдется. Область, в которой заданные исходные значения сходятся к решению, называют *областью сходимости*. Если исходные значения лежат за пределами этой области, то решение получить не удастся. Условия, при которых метод итерации сходится к решению, математически задаются системой неравенств

$$\left\{ \begin{array}{l} \left| \frac{\partial g_1}{\partial x_1} \right| + \left| \frac{\partial g_1}{\partial x_2} \right| + \dots + \left| \frac{\partial g_1}{\partial x_n} \right| < 1, \\ \left| \frac{\partial g_2}{\partial x_1} \right| + \left| \frac{\partial g_2}{\partial x_2} \right| + \dots + \left| \frac{\partial g_2}{\partial x_n} \right| < 1, \\ \dots\dots\dots \\ \left| \frac{\partial g_n}{\partial x_1} \right| + \left| \frac{\partial g_n}{\partial x_2} \right| + \dots + \left| \frac{\partial g_n}{\partial x_n} \right| < 1. \end{array} \right. \quad (14)$$

С увеличением числа уравнений и, следовательно, числа переменных в нелинейной системе область сходимости уменьшается и в случае очень больших систем сходимость обеспечивается лишь при условии, что исходные значения переменных очень близки к истинному решению.

Пример 5. Методом итераций приближенно найти решение системы уравнений

$$\begin{cases} 3x_1 - \cos(x_2) = 0,9; \\ \sin(x_1 - 0,6) - x_2 = 1,6. \end{cases}$$

Перепишем данную систему в виде

$$\begin{cases} x_1 = \frac{1}{3} \cos(x_2) + 0,3 ; \\ x_2 = \sin(x_1 - 0,6) - 1,6. \end{cases}$$

Выбор начального приближения произведем графически (рис. 11). Из графика видим, что система имеет одно решение, заключенное в области D :

$$0 < x_1 < 0,3; \quad -2,2 < x_2 < -1,8.$$

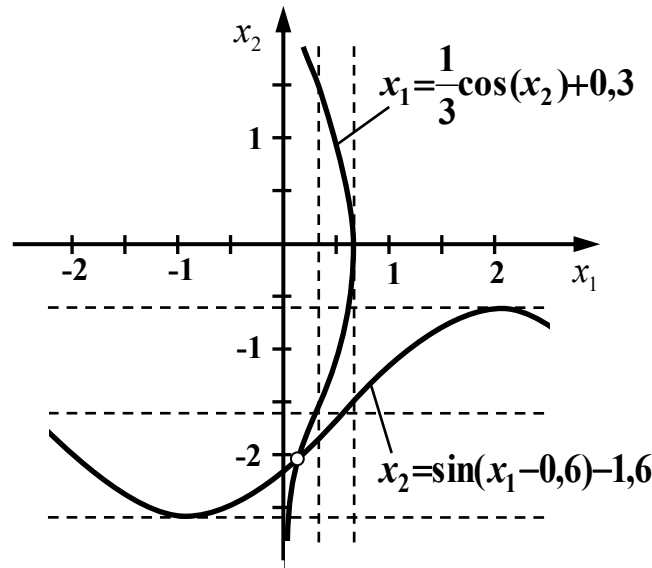


Рис. 11. Выбор начального приближения

Чтобы убедиться в том, что метод итераций применим для уточнения решения данной системы, проверим для нее условия сходимости (14).

С этой целью вычислим первые производные от правых частей уравнений системы

$$\begin{cases} x_1 = g_1(x_1, x_2) = \frac{1}{3} \cos(x_2) + 0,3; \\ x_2 = g_2(x_1, x_2) = \sin(x_1 - 0,6) - 1,6. \end{cases}$$

Так как

$$\begin{aligned} \frac{\partial g_1}{\partial x_1} &= 0, & \frac{\partial g_1}{\partial x_2} &= -\frac{1}{3} \sin(x_2), \\ \frac{\partial g_2}{\partial x_1} &= \cos(x_1 - 0,6), & \frac{\partial g_2}{\partial x_2} &= 0, \end{aligned}$$

то в области D имеем

$$\begin{aligned} \left| \frac{\partial g_1}{\partial x_1} \right| + \left| \frac{\partial g_1}{\partial x_2} \right| &= \left| -\frac{1}{3} \sin(x_2) \right| = \left| -\frac{1}{3} \sin(-1,8) \right| = 0,3246 < 1; \\ \left| \frac{\partial g_2}{\partial x_1} \right| + \left| \frac{\partial g_2}{\partial x_2} \right| &= |\cos(x_1 - 0,6)| = |\cos(0,3 - 0,6)| = 0,9553 < 1. \end{aligned}$$

Таким образом, условия сходимости выполняются. За начальные приближения принимаем $x_1^{(0)} = 0,15$, $x_2^{(0)} = -2$.

После первой итерации получим значения

$$\begin{cases} x_1^{(1)} = \frac{1}{3} \cos(-2) + 0,3 = 0,1613; \\ x_2^{(1)} = \sin(0,1613 - 0,6) - 1,6 = -2,0248. \end{cases}$$

Аналогично находятся дальнейшие приближения. Результаты вычислений приведены в табл. 2.

Таблица 2

Приближения корней методом итераций

k	$x_1^{(k)}$	$x_2^{(k)}$
0	0,3	-2
1	0,1613	-2,0248
2	0,1538	-2,0315
3	0,1518	-2,0333
4	0,1513	-2,0338
5	0,1511	-2,0340

Остановившись на приближении $\mathbf{x}^{(5)}$, будем иметь: $x_1 = 0,1511$, $x_2 = -2,034$.

7.3. ЗАДАНИЯ К РАБОТЕ

Решить систему нелинейных уравнений вида

$$\begin{cases} f_1(x_1, x_2) = 0, \\ f_2(x_1, x_2) = 0. \end{cases}$$

с точностью $\varepsilon = 0,0001$ используя метод простой итерации.

Варианты заданий

№ варианта	Система нелинейных уравнений	№ варианта	Система нелинейных уравнений
1	$\begin{cases} \sin x_1 + 2x_2 - 2 = 0, \\ \cos(x_2 - 1) + x_1 - 0,7 = 0. \end{cases}$	2	$\begin{cases} \sin x_2 + x_1 + 0,4 = 0, \\ 2x_2 - \cos(x_1 + 1) = 0. \end{cases}$
3	$\begin{cases} \sin(x_1 + 0,5) - x_2 - 1 = 0, \\ \cos(x_2 - 2) + x_1 = 0. \end{cases}$	4	$\begin{cases} \sin(x_1 + 2) - x_2 - 1,5 = 0, \\ \cos(x_2 - 2) + x_1 - 0,5 = 0. \end{cases}$
5	$\begin{cases} \sin(x_1 + 1) - x_2 - 1,2 = 0, \\ 2x_1 - \cos x_2 - 2 = 0. \end{cases}$	6	$\begin{cases} \sin(x_1 + 0,5) - x_2 - 1 = 0, \\ x_1 + \cos(x_2 - 2) = 0. \end{cases}$
7	$\begin{cases} \cos x_1 + x_2 - 1,5 = 0, \\ 2x_1 - \sin(x_2 - 0,5) - 1 = 0. \end{cases}$	8	$\begin{cases} \cos(x_1 + 0,5) - x_2 - 2 = 0, \\ \sin x_2 - 2x_1 - 1 = 0. \end{cases}$

9	$\begin{cases} \cos(x_1 + 0,5) + x_2 - 0,8 = 0, \\ \sin x_2 - 2x_1 - 1,6 = 0. \end{cases}$	10	$\begin{cases} \cos(x_1 - 1) + x_2 - 0,5 = 0, \\ x_1 - \cos x_2 - 3 = 0. \end{cases}$
11	$\begin{cases} \sin(x_1 - 1) + x_2 - 1,3 = 0, \\ x_1 - \sin(x_2 + 1) - 0,8 = 0. \end{cases}$	12	$\begin{cases} \sin(x_2 + 1) - x_1 - 1,2 = 0, \\ 2x_2 + \cos x_1 - 2 = 0. \end{cases}$
13	$\begin{cases} 2x_2 - \cos(x_1 + 1) = 0, \\ x_1 + \sin x_2 + 0,4 = 0. \end{cases}$	14	$\begin{cases} \cos(x_2 - 1) + x_1 - 0,5 = 0, \\ x_2 - \cos x_1 - 3 = 0. \end{cases}$
15	$\begin{cases} \sin x_2 + 2x_1 - 2 = 0, \\ \cos(x_1 - 1) + x_2 - 0,7 = 0. \end{cases}$	16	$\begin{cases} \cos x_2 + x_1 - 1,5 = 0, \\ 2x_2 - \sin(x_1 - 0,5) - 1 = 0. \end{cases}$
17	$\begin{cases} \sin(x_2 + 0,5) - x_1 - 1 = 0, \\ \cos(x_1 - 2) + x_2 = 0. \end{cases}$	18	$\begin{cases} \cos(x_2 + 0,5) + x_1 - 0,8 = 0, \\ \sin x_1 - 2x_2 - 1,6 = 0. \end{cases}$
19	$\begin{cases} \sin(x_2 - 1) + x_1 - 1,3 = 0, \\ x_2 - \sin(x_1 + 1) - 0,8 = 0. \end{cases}$	20	$\begin{cases} 2x_1 - \cos(x_2 + 1) = 0, \\ x_2 + \sin x_1 + 0,4 = 0. \end{cases}$
21	$\begin{cases} \cos(x_2 + 1,5) - x_1 - 2 = 0, \\ \sin x_1 - 2x_2 - 1 = 0. \end{cases}$	22	$\begin{cases} \sin(x_2 + 2) - x_1 - 1,5 = 0, \\ x_2 + \cos(x_1 - 2) - 0,5 = 0. \end{cases}$
23	$\begin{cases} \cos(x_1 - 2) + x_2 - 0,8 = 0, \\ x_1 - \cos x_2 - 1 = 0. \end{cases}$	24	$\begin{cases} \sin x_1 + 0,5x_2 - 1,2 = 0, \\ \cos(x_2 - 1) + x_1 - 1 = 0. \end{cases}$
25	$\begin{cases} \cos x_1 + 2x_2 - 1,2 = 0, \\ 2x_1 - \sin(x_2 - 0,5) - 2 = 0. \end{cases}$	26	$\begin{cases} \sin(x_1 + 0,5) - 2x_2 - 1,4 = 0, \\ \cos(x_2 - 2) + 2x_1 = 0. \end{cases}$
27	$\begin{cases} \cos(x_1 + 0,5) + x_2 - 1 = 0, \\ \sin x_2 - 2x_1 - 2,6 = 0. \end{cases}$	28	$\begin{cases} \sin(x_1 - 1) + x_2 - 1,5 = 0, \\ x_1 - \sin(x_2 + 2) - 1 = 0. \end{cases}$
29	$\begin{cases} \sin(x_2 + 1) - 2x_1 - 1 = 0, \\ 2x_2 + \cos x_2 - 2 = 0. \end{cases}$	30	$\begin{cases} \cos(x_2 - 1) - x_1 - 0,8 = 0, \\ x_2 + \cos x_1 - 1,5 = 0. \end{cases}$

7.4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теорией приближенного решения систем нелинейных уравнений.
2. Преобразовать исходную систему уравнений к виду, пригодному для метода простых итераций (см. пример 5).
3. Построить графики уравнений, определить примерную область решения.
4. Проверить условия сходимости метода итераций для преобразованной системы (см. пример 5).
5. На основании графиков и условий сходимости выбрать начальные приближения для неизвестных системы.

6. Используя программу 4 из приложения 1, решить на компьютере заданную систему нелинейных уравнений с точностью ε методом простых итераций.
7. Оформить отчет по лабораторной работе.

7.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлении студентом аккуратно оформленного отчета и при умении применять на практике численный метод решения системы нелинейных уравнений.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием лабораторной работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;
- графический материал;
- схему алгоритма применяемого метода;
- текст используемой программы;
- результаты вычислительного эксперимента;
- анализ полученных результатов.

7.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Постановка задачи решения системы нелинейных уравнений. Основные этапы решения задачи.
2. Метод простой итерации: описание метода; условие сходимости; критерий окончания; приведение системы к виду, удобному для итераций.
3. Что называют областью сходимости метода итераций?
4. От чего зависит количество итераций в процессе решения?
5. Недостатки метода простой итерации.

8. РАБОТА №8. РЕШЕНИЕ СИСТЕМЫ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

8.1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение студентами умения решения на компьютере системы линейных алгебраических уравнений прямым методом.

8.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

8.2.1. Общие сведения

Способы решения систем линейных уравнений делятся на две группы:

- 1) *прямые методы*, представляющие собой алгоритмы для вычисления корней системы (решение систем с помощью обратной матрицы, правило Крамера, метод Гаусса и др.);
- 2) *итерационные методы*, позволяющие получить решение системы с заданной точностью путем сходящихся итерационных процессов (метод итерации, метод Зейделя и др.).

Прямые методы, как правило, имеют высокую эффективность, но в случае больших матриц они уступают итерационным.

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных $x_1, x_2, \dots, x_n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases} \quad (15)$$

В соответствии с правилом умножения матриц рассмотренная система линейных уравнений может быть записана в матричном виде (*матричным уравнением* называется уравнение, коэффициенты и неизвестные которого – прямоугольные матрицы соответствующей размерности)

$$Ax = b, \quad (16)$$

где

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}. \quad (17)$$

$$\begin{bmatrix} 1 & 0 & \dots & 0 & x_1 \\ 0 & 1 & \dots & 0 & x_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & x_n \end{bmatrix}.$$

Последний $(n + 1)$ столбец этой матрицы содержит решение системы (15).

Вычислительная процедура реализации данного метода строится следующим образом.

Сначала нормируют первое уравнение, для этого его коэффициенты делят на a_{11} . Затем первое уравнение умножают на первые коэффициенты $a_{i,1}$ всех других уравнений и последовательно вычитают из остальных уравнений. В результате первая переменная будет исключена из всех уравнений, кроме первого. На следующем этапе (шаге) решения такая же процедура применяется к остальным $n - 1$ уравнениям. В результате из оставшихся $n - 2$ уравнений исключается вторая неизвестная. Всю процедуру повторяют до тех пор, пока после n шагов вся система не будет приведена к треугольному виду. Математически алгоритм прямого хода можно описать так.

На k -м шаге процесса исключения новые нормированные коэффициенты k -го уравнения (его называют ведущим) имеют вид

$$a_{kj}^* = \frac{a_{kj}}{a_{kk}} \quad \text{и} \quad b_k^* = \frac{b_k}{a_{kk}}, \quad (j = 1, 2, \dots, n),$$

а новые коэффициенты в следующих уравнениях принимают вид

$$a_{ij}^* = a_{ij} - a_{ik} \cdot a_{kj}^* \\ \text{и} \quad b_i^* = b_i - a_{ik} \cdot b_k^*, \quad (i = k+1, 2, \dots, n, \quad j = 1, 2, \dots, n).$$

Выполняя эту процедуру, следует помнить, что коэффициенты нижестоящих уравнений a_{ij} и b_i меняются на каждом шаге. Например, коэффициенты a_{ij}^* на следующем шаге превращаются в коэффициенты a_{ij} .

Значения неизвестных x_i вычисляются на обратном ходе по формулам (19).

Применение описанного метода усложняется, если диагональный коэффициент ведущего уравнения равен нулю. В этом случае ведущее уравнение нельзя нормировать. Однако, изменив порядок, в котором расположены уравнения системы, эту трудность можно обойти. Можно показать, что наименьшая погрешность округления достигается тогда, когда диагональный коэффициент (ведущий элемент) имеет наибольшее значение. Поэтому строку с нулевым или малым ведущим элементом надо заменить на ту из стоящих под ней строк, в которой в том же столбце стоит элемент, имеющий наибольшее значение. Такая процедура называется методом Гаусса с выбором ведущего элемента.

Схема алгоритма решения системы уравнений методом Гаусса приведена на рис. 12.

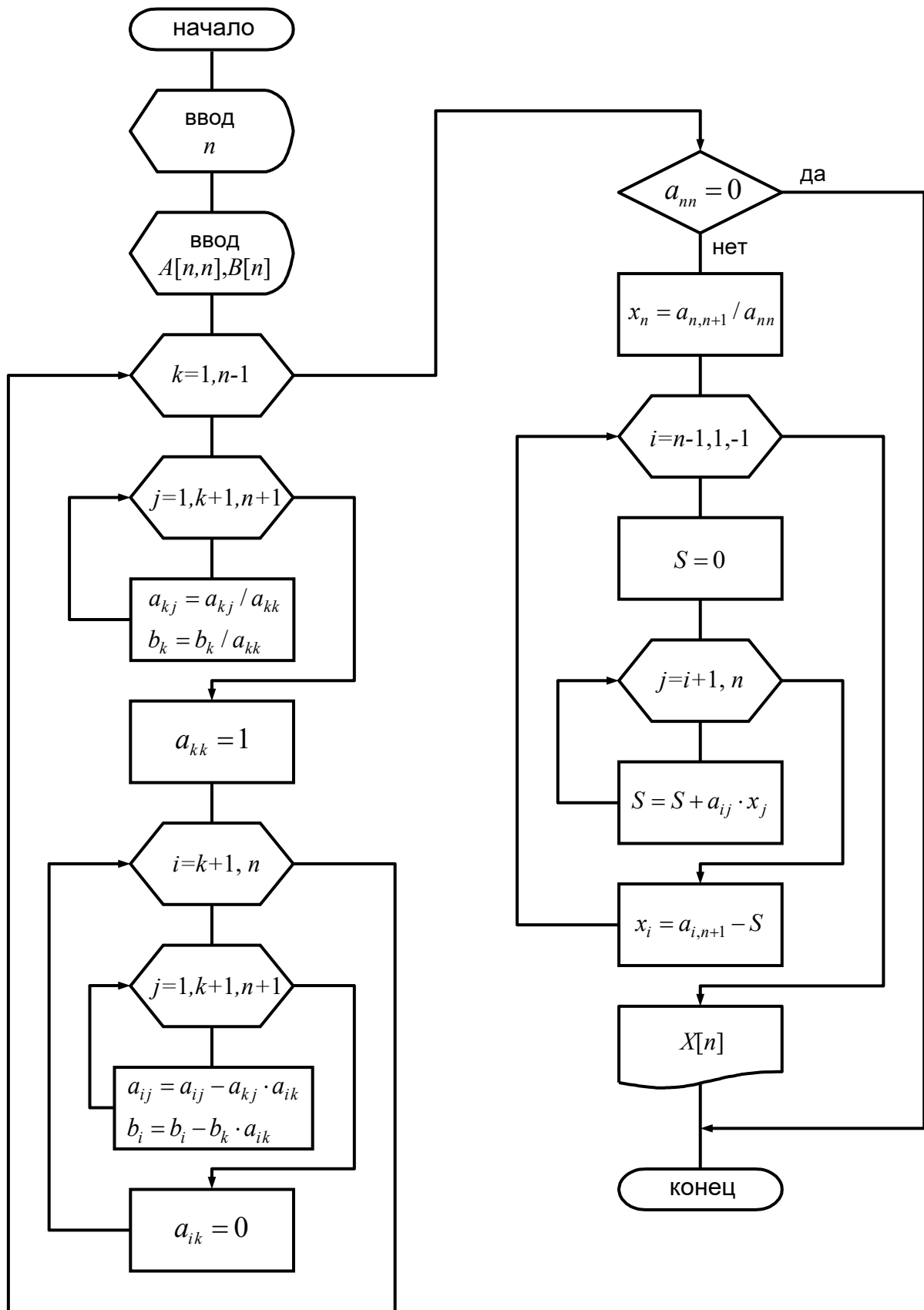


Рис. 12. Алгоритм метода исключения неизвестных Гаусса

Пример 6. Методом исключения Гаусса решить систему уравнений

$$\begin{cases} x_1 + x_2 + x_3 - x_4 = 2, \\ x_1 - x_2 - x_3 + x_4 = 0, \\ 2x_1 + x_2 - x_3 + 2x_4 = 9, \\ 3x_1 + x_2 + 2x_3 - x_4 = 7. \end{cases}$$

Для удобства обозначим уравнения буквами и будем выписывать только коэффициенты при неизвестных и свободные члены уравнений. Тогда исходная система примет вид

$$\begin{cases} A_1 \\ A_2 \\ A_3 \\ A_4 \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 2 & 1 & -1 & 2 \\ 3 & 1 & 2 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 9 \\ 7 \end{bmatrix}.$$

Исключая из уравнений члены, содержащие x_1 , получим

$$\begin{cases} B_1 = A_1 / 1 \\ B_2 = A_2 - B_1 \\ B_3 = A_3 - 2B_1 \\ B_4 = A_4 - 3B_1 \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & -1 \\ 0 & -2 & -2 & 2 \\ 0 & -1 & -3 & 4 \\ 0 & -2 & -1 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \\ 5 \\ 1 \end{bmatrix}.$$

После исключения членов с x_2 имеем

$$\begin{cases} B_1 \\ C_2 = B_2 / (-2) \\ C_3 = B_3 + C_2 \\ C_4 = B_4 + 2C_2 \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & -2 & 3 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 6 \\ 3 \end{bmatrix}.$$

После исключения членов с x_3 имеем

$$\begin{cases} B_1 \\ C_2 \\ D_3 = C_3 / (-2) \\ D_4 = C_4 - D_3 \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -3/2 \\ 0 & 0 & 0 & 3/2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ -3 \\ 6 \end{bmatrix}.$$

Нормируя последнее уравнение, получим

$$\left\{ \begin{array}{l} B_1 \\ C_2 \\ D_3 \\ E_4 = 2D_4/3 \end{array} \right. \Rightarrow \left[\begin{array}{cccc} 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -3/2 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} 2 \\ 1 \\ -3 \\ 4 \end{array} \right].$$

Возвращаясь к форме уравнений, получим

$$\left\{ \begin{array}{l} x_1 + x_2 + x_3 - x_4 = 2, \\ x_2 + x_3 - x_4 = 1, \\ x_3 - \frac{3}{2}x_4 = -3, \\ x_4 = 7. \end{array} \right.$$

Далее выполняем обратный ход. Значение $x_4=4$ подставляем в третье уравнение системы и находим x_3 . Во второе уравнение подставляем x_4 , x_3 и находим x_2 , а в первое x_4 , x_3 , x_2 и находим x_1 . Решением системы будут значения:

$$x_1=1, \quad x_2=2, \quad x_3=3, \quad x_4=4.$$

8.3. ЗАДАНИЯ К РАБОТЕ

Дана система линейных алгебраических уравнений относительно неизвестных $x_1, x_2, \dots, x_n$.

Решить эту систему на компьютере методом последовательных исключений неизвестных Гаусса.

Варианты заданий

№ варианта	Система линейных уравнений	№ варианта	Система линейных уравнений
1	$\begin{cases} 2x_1 + x_2 + 2x_3 + 3x_4 = 8 \\ 3x_1 + 3x_3 = 6 \\ 2x_1 - x_2 + 3x_4 = 4 \\ x_1 + 2x_2 - x_3 + 2x_4 = 4 \end{cases}$	2	$\begin{cases} 2x_1 + x_2 - 5x_3 + x_4 = -4 \\ x_1 - 3x_2 - 6x_4 = -7 \\ 2x_2 - x_3 + 2x_4 = 2 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -2 \end{cases}$
3	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 26 \\ 2x_1 + 3x_2 + 4x_3 + x_4 = 34 \\ 3x_1 + 4x_2 + x_3 + 2x_4 = 26 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$	4	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 22 \\ 2x_1 + 3x_2 + x_3 + 2x_4 = 17 \\ x_1 + x_2 + x_3 - x_4 = 8 \\ x_1 - 2x_3 - 3x_4 = -7 \end{cases}$

№ варианта	Система линейных уравнений	№ варианта	Система линейных уравнений
5	$\begin{cases} 9x_1 + 10x_2 - 7x_3 - x_4 = 23 \\ 7x_1 - x_3 - 5x_4 = 37 \\ 5x_1 - 2x_3 + x_4 = 22 \\ 4x_1 + x_2 + 2x_3 + 3x_4 = 26 \end{cases}$	6	$\begin{cases} 2x_1 - 8x_2 - 3x_3 - 2x_4 = -18 \\ x_1 - 2x_2 + 3x_3 - 2x_4 = 28 \\ x_2 + x_3 + x_4 = 10 \\ 11x_2 + x_3 + 2x_4 = 21 \end{cases}$
7	$\begin{cases} x_1 + 2x_2 + x_3 + 2x_4 = 8 \\ 3x_1 - 5x_2 + 3x_3 + x_4 = 3 \\ 2x_1 + 7x_2 - x_3 - 5x_4 = -2 \\ 2x_1 - 3x_2 - 2x_3 + 3x_4 = 3 \end{cases}$	8	$\begin{cases} 7x_1 + 2x_2 + 3x_3 + x_4 = 16 \\ 5x_1 - 3x_2 + 2x_3 - 5x_4 = 10 \\ 10x_1 - 11x_2 + 5x_3 - 8x_4 = 28 \\ 4x_1 - 3x_2 + 2x_3 + 5x_4 = 18 \end{cases}$
9	$\begin{cases} x_1 - 2x_2 - 8x_4 = -7 \\ x_1 + 4x_2 - 7x_3 + 6x_4 = -8 \\ x_1 + x_2 - 5x_3 + x_4 = -10 \\ 2x_1 - x_2 + 2x_4 = 7 \end{cases}$	10	$\begin{cases} 7x_1 + 7x_2 - 7x_3 - 2x_4 = 5 \\ 3x_1 + 4x_2 + 5x_3 + 8x_4 = 60 \\ 2x_1 + 2x_2 + 2x_3 + x_4 = 27 \\ 2x_1 - 2x_3 - x_4 = -1 \end{cases}$
11	$\begin{cases} 2x_1 + 2x_2 + 6x_3 + x_4 = 15 \\ -x_2 + 2x_3 + x_4 = 18 \\ 4x_1 - 3x_2 + x_3 - 5x_4 = 37 \\ 3x_1 - 5x_2 + x_3 - x_4 = 30 \end{cases}$	12	$\begin{cases} 6x_1 - 9x_2 + 5x_3 + x_4 = 124 \\ 7x_2 - 5x_3 - x_4 = -54 \\ 5x_1 - 5x_2 + 2x_3 + 4x_4 = 83 \\ 3x_1 - 9x_2 + x_3 + 6x_4 = 45 \end{cases}$
13	$\begin{cases} 4x_1 - 5x_2 + 7x_3 + 5x_4 = 165 \\ 2x_1 + x_2 - 3x_3 - x_4 = -15 \\ 9x_1 + 4x_3 - x_4 = 194 \\ x_1 - x_2 - 2x_3 - 3x_4 = -19 \end{cases}$	14	$\begin{cases} 3x_1 + 12x_2 - x_3 = 18 \\ -5x_1 + 2x_2 + 32x_4 = -15 \\ 2x_1 + 16x_3 - 3x_4 = 0 \\ 12x_1 + 3x_2 = 21 \end{cases}$
15	$\begin{cases} 4x_1 + 2x_2 + 32x_3 = -19 \\ 2x_1 + 30x_2 - 4x_4 = 39 \\ 36x_1 + 4x_3 - 5x_4 = 40 \\ 11x_3 + 40x_4 = 31 \end{cases}$	16	$\begin{cases} x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ 3x_1 - x_2 - x_3 - 2x_4 = -4 \\ 2x_1 + 3x_2 - x_3 - x_4 = -6 \\ x_1 + 2x_2 + 3x_3 - x_4 = -4 \end{cases}$
17	$\begin{cases} x_1 + 2x_2 + 3x_3 - 2x_4 = 6 \\ 2x_1 - x_2 - 2x_3 - 3x_4 = 8 \\ 3x_1 + 2x_2 - x_3 + 2x_4 = 4 \\ 2x_1 - 3x_2 + 2x_3 + x_4 = -8 \end{cases}$	18	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 5 \\ 2x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ 3x_1 + 2x_2 + x_3 + 2x_4 = 1 \\ 4x_1 + 3x_2 + 2x_3 + x_4 = -5 \end{cases}$

№ варианта	Система линейных уравнений	№ варианта	Система линейных уравнений
19	$\begin{cases} x_2 - 3x_3 + 4x_4 = -5 \\ x_1 - 2x_3 + 3x_4 = -4 \\ 3x_1 + 2x_2 - 5x_4 = 12 \\ 4x_1 + 3x_2 - 5x_3 = 5 \end{cases}$	20	$\begin{cases} x_1 - 2x_2 + x_3 + x_4 = 1 \\ x_1 - 2x_2 + x_3 - x_4 = -1 \\ x_1 - 2x_2 + x_3 + 5x_4 = 5 \\ x_1 + x_2 + x_3 + x_4 = 0 \end{cases}$
21	$\begin{cases} 2x_1 + x_2 - x_3 + x_4 = 1 \\ 3x_1 - 2x_2 + 2x_3 - 3x_4 = 2 \\ 5x_1 + x_2 - x_3 + 2x_4 = -1 \\ 2x_1 - x_2 - x_3 - 3x_4 = 4 \end{cases}$	22	$\begin{cases} x_1 - 2x_2 + 3x_3 - 4x_4 = -4 \\ x_2 - x_3 + x_4 = -3 \\ x_1 + 3x_2 - 3x_4 = 1 \\ -7x_2 + 3x_3 + x_4 = -3 \end{cases}$
23	$\begin{cases} 2x_1 - 4x_2 + 9x_3 - 2x_4 = 26 \\ 7x_1 + 3x_2 - 6x_3 + 8x_4 = 7 \\ 7x_1 + 9x_2 - 9x_3 + 4x_4 = 9 \\ x_1 - 3x_2 + 5x_3 - x_4 = 12 \end{cases}$	24	$\begin{cases} 4x_1 + 20x_2 + x_3 = 24 \\ 16x_1 + 2x_2 - 2x_4 = -13 \\ -4x_1 + 4x_3 + 32x_4 = 0 \\ 2x_1 + 10x_3 = 7 \end{cases}$
25	$\begin{cases} 6x_1 - x_2 + 10x_3 - x_4 = 158 \\ 2x_1 + x_2 + 10x_3 + 7x_4 = 128 \\ 3x_1 - 2x_2 - 2x_3 - x_4 = 7 \\ x_1 - 12x_2 + 2x_3 - x_4 = 17 \end{cases}$	26	$\begin{cases} 2x_1 - x_2 + 4x_3 + x_4 = 66 \\ 2x_2 - 6x_3 + x_4 = -63 \\ 8x_1 - 3x_2 + 6x_3 - 5x_4 = 146 \\ 2x_1 - 7x_2 + 6x_3 - x_4 = 80 \end{cases}$
27	$\begin{cases} x_1 + 2x_2 + x_3 - x_4 = 3 \\ 3x_1 - 5x_2 + 3x_3 + 2x_4 = 3 \\ 2x_1 + 7x_2 - x_3 - 3x_4 = 5 \\ x_1 + x_2 - x_3 + x_4 = 2 \end{cases}$	28	$\begin{cases} x_1 - 2x_2 + 6x_3 + x_4 = 88 \\ 5x_1 + 2x_3 - 3x_4 = 88 \\ 7x_1 - 3x_2 + 7x_3 + 2x_4 = 181 \\ 3x_1 - 7x_2 + 5x_3 + 2x_4 = 99 \end{cases}$
29	$\begin{cases} 2x_1 - 3x_3 - 2x_4 = -16 \\ 2x_1 - x_2 + 13x_3 + 4x_4 = 213 \\ 3x_1 + x_2 + 2x_3 + x_4 = 72 \\ x_1 - 12x_3 - 5x_4 = -159 \end{cases}$	30	$\begin{cases} 7x_1 + 2x_2 + 3x_3 - 2x_4 = 11 \\ 5x_1 - 3x_2 + 2x_3 + x_4 = 16 \\ 10x_1 - 11x_2 + 5x_3 - 3x_4 = 30 \\ 2x_1 + 5x_2 - x_3 + 4x_4 = 6 \end{cases}$

8.4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теорией решения систем линейных алгебраических уравнений.
2. Решить заданную систему уравнений методом Гаусса вручную (см. пример 6).
3. Используя программу 5 из приложения 1, решить заданную систему уравнений методом исключений Гаусса на компьютере.
4. Оформить отчет по лабораторной работе.

8.5. ОТЧЕТНОСТЬ

Работа засчитывается по представлению студентом аккуратно оформленного отчета и при умении применять на практике прямой метод решения системы линейных алгебраических уравнений.

Отчет выполняется на компьютере и должен содержать:

- титульный лист с названием лабораторной работы;
- цель работы;
- постановку задачи;
- необходимый теоретический материал;
- решение заданной системы уравнений вручную;
- схему алгоритма применяемого метода;
- текст используемой программы;
- результаты расчета на компьютере.

8.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Постановка задачи решения системы линейных алгебраических уравнений. Условие единственности решения.
2. Какие методы применяются для решения систем линейных уравнений?
3. Метод Гаусса (схема единственного деления): последовательность действий, прямой и обратный ход.
4. Чем отличается метод Гаусса с выбором главного элемента? В чем его преимущество?

9. РАБОТА №9. ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ ПРОСТЫХ ИНТЕГРАЛОВ

9.1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение студентами умения приближенного вычисления определенных интегралов.

9.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

9.2.1. Общие сведения

В классической математике определенный интеграл вычисляется через первообразную подынтегральной функции по формуле Ньютона-Лейбница

$$\int_a^b f(x)dx = F(b) - F(a).$$

Эта формула не удобна для вычислений на компьютере, поскольку процедура получения первообразной $F(x)$ может быть сложной. Может случиться, что первообразная $F(x)$ не выражается через элементарные функции, либо вычисляется слишком сложно. В этих случаях прибегают к методам приближенного интегрирования, позволяющим найти значение определенного интеграла с любой степенью точности.

Приближенные методы основаны на геометрическом смысле. С геометрической точки зрения значение определенного интеграла есть площадь S фигуры, ограниченной кривой $f(x)$, прямыми $x = a$, $x = b$ и осью абсцисс (рис. 13).

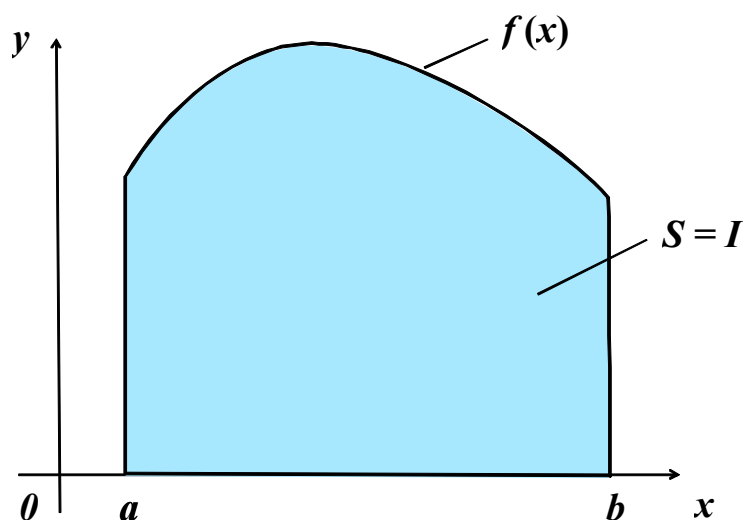


Рис. 13. Геометрический смысл определенного интеграла

9.2.2. Метод прямоугольников

Геометрически данный метод означает, что площадь исходной фигуры (криволинейной трапеции) приближенно заменяется на сумму площадей прямоугольников.

$$\int_a^b f(x)dx \approx \frac{b-a}{n}(y_0 + y_1 + \dots + y_{n-1}) = h(y_0 + y_1 + \dots + y_{n-1}). \quad (20)$$

Правая часть формулы (20) - это площадь ступенчатой фигуры, заштрихованной на рис. 14. Аналогично записывается формула правых прямоугольников:

$$\int_a^b f(x)dx \approx \frac{b-a}{n}(y_1 + y_2 + \dots + y_n) = h(y_1 + y_2 + \dots + y_n). \quad (20')$$

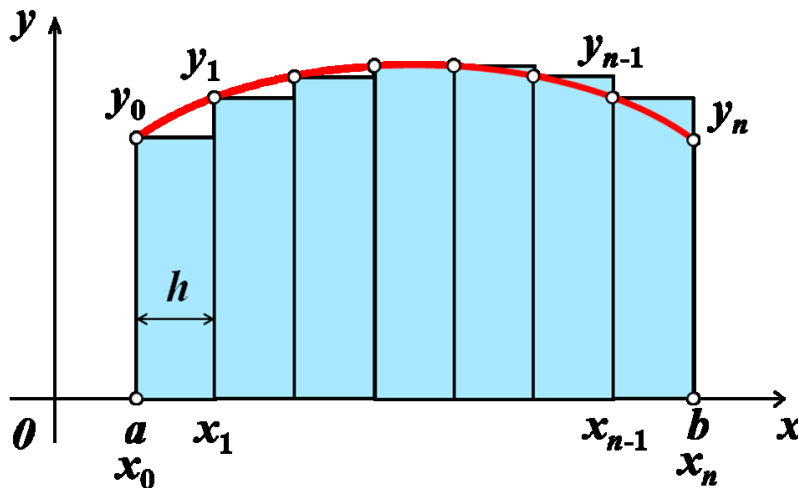


Рис. 14. Геометрический смысл формулы левых прямоугольников

9.2.3. Метод трапеций

Увеличение точности формул (19) и (20') достигается увеличением числа n отрезков разбиения промежутка интегрирования, но это ведет к увеличению количества вычислений. Чтобы получить более точный результат без увеличения числа n , применяют метод трапеций.

Если найти среднее арифметическое правых частей формул (20) и (20'), то получим

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=0}^{n-1} \frac{y_i + y_{i+1}}{2}. \quad (21)$$

Геометрический смысл ее таков: если на каждом отрезке разбиения дугу графика подынтегральной функции $y = f(x)$ заменить стягивающей ее хордой,

то мы получим прямоугольную трапецию и формула (21) представляет собой площадь фигуры, состоящей из таких трапеций (рис. 15).

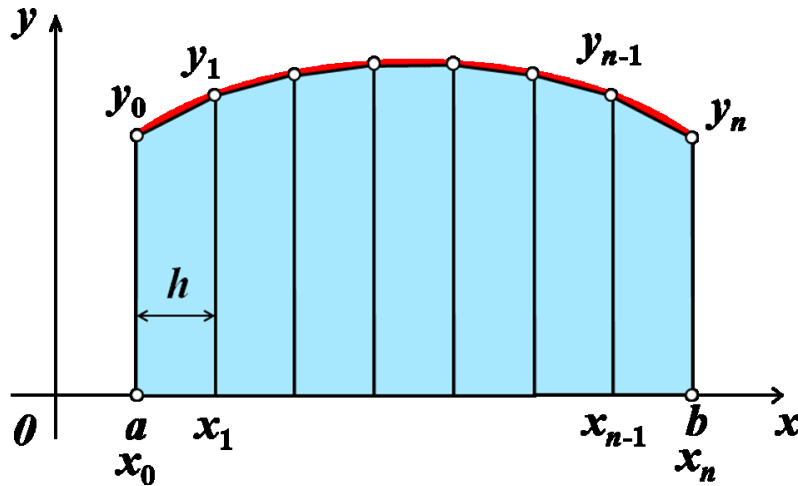


Рис. 15. Геометрический смысл формулы трапеций

Приведя в формуле (21) подобные члены, получим формулу трапеций:

$$\int_a^b f(x) dx \approx \frac{b-a}{n} \left(\frac{1}{2} y_0 + y_1 + y_2 + \dots + y_{n-1} + \frac{1}{2} y_n \right) = h \cdot \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right). \quad (22)$$

Если обозначить через R_n разность между левой и правой частями формулы (22), то будем иметь

$$\int_a^b f(x) dx = \frac{b-a}{n} \left(\frac{1}{2} y_0 + y_1 + y_2 + \dots + y_{n-1} + \frac{1}{2} y_n \right) + R_n.$$

Величину R_n называют поправочным членом, или погрешностью формулы (22). Если подынтегральная функция $f(x)$ имеет на отрезке $[a, b]$ непрерывную вторую производную, то для погрешности справедлива оценка

$$|R_n| \leq \frac{(b-a)^3}{12n^2} \cdot M_2, \quad (23)$$

где M_2 - максимум модуля второй производной подынтегральной функции на отрезке $[a, b]$.

9.2.4. Метод параболических трапеций (метод Симпсона)

Разделим отрезок $[a, b]$ на $2n$ равных частей (рис. 16). Точки деления будут $x_0 = a, x_1, x_2, \dots, x_{2n-2}, x_{2n-1}, x_{2n} = b$. Проведем параболу $y = Ax^2 + Bx + C$ че-

рез точки (x_0, y_0) , (x_1, y_1) и (x_2, y_2) . Такую параболу всегда можно провести через три точки и притом единственным образом.

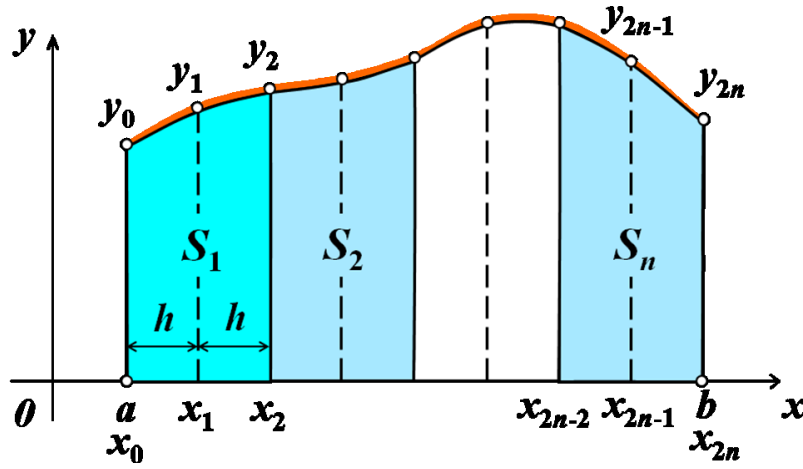


Рис. 16. Геометрический смысл формулы Симпсона

Вычислим площадь криволинейной трапеции под параболой. Если обозначить шаг разбиения промежутка через h ,

$$h = \frac{(b-a)}{2n},$$

то в соответствии с известной из геометрии формулой имеем

$$\int_{x_0}^{x_2} f(x) dx \approx S_1 = \frac{h}{3} (y_0 + 4y_1 + y_2).$$

Аналогично проводя параболу через точки x_2, x_3, x_4 и т.д., получим

$$\int_{x_2}^{x_4} f(x) dx \approx S_2 = \frac{h}{3} (y_2 + 4y_3 + y_4),$$

...

$$\int_{x_{2n-2}}^{x_{2n}} f(x) dx \approx S_n = \frac{h}{3} (y_{2n-2} + 4y_{2n-1} + y_{2n}).$$

Складывая эти формулы, окончательно получим

$$\begin{aligned} \int_a^b f(x) dx &\approx S_1 + S_2 + \dots + S_n = \\ &= \frac{h}{3} [(y_0 + y_{2n}) + 2(y_2 + y_4 + \dots + y_{2n-2}) + 4(y_1 + y_3 + \dots + y_{2n-1})] = \\ &= \frac{h}{3} \left[(y_0 + y_{2n}) + 2 \sum_{i=2,4,\dots}^{2n-2} y_i + 4 \sum_{i=1,3,\dots}^{2n-1} y_i \right]. \end{aligned} \quad (24)$$

Эта формула носит название формулы Симпсона или параболических трапеций. Геометрически этот метод заключается в том, что площадь криволинейной трапеции приближенно заменяется суммой площадей параболических трапеций.

Если подынтегральная функция $y=f(x)$ имеет на отрезке $[a, b]$ непрерывную четвертую производную, то для погрешности или поправочного члена формулы (24) имеет место оценка

$$|R_n| \leq \frac{(b-a)^5}{180(2n)^4} \cdot M_4, \quad (25)$$

где M_4 - максимум модуля четвертой производной подынтегральной функции на отрезке $[a, b]$.

Сравнивая между собой оценки (23) и (25), замечаем, что с увеличением n поправочный член формулы трапеций уменьшается пропорционально величине $1/n^2$, а для формулы Симпсона - пропорционально $1/n^4$, т.е. метод Симпсона дает более точные результаты.

Погрешность вычисления интеграла во всех рассмотренных методах зависит от выбранного шага, т.е. числа разбиений n . Можно предложить алгоритм, позволяющий автоматически достигать желаемой погрешности (см. рис. 17). Последовательно выполняются несколько расчетов значения интеграла, начиная с небольшого начального числа разбиений, и с каждым последующим расчетом число разбиений увеличивается (повышается точность), например в два раза. Если при увеличении числа разбиений вдвое модуль разности соответствующих значений интеграла меньше требуемой точности, то требуемая точность достигнута и n больше не увеличивается.

Условие достижения заданной погрешности будет следующим

$$|I_n - I_{2n}| \leq \varepsilon,$$

где I_n - значение интеграла при числе разбиений n ;
 I_{2n} - значение интеграла при числе разбиений $2n$;
 ε - заданная (желаемая) погрешность.

9.3. РЕШЕНИЕ ПРИМЕРОВ

Пример 7. Методом трапеций вычислить интеграл

$$\int_0^1 \frac{dx}{1+x^2}$$

с погрешностью $\varepsilon = 0,005$.

Для выбора шага, с которым будем вычислять таблицу значений подынтегральной функции, оценим величину погрешности по формуле (23).

$$\text{Имеем } f(x) = \frac{1}{1+x^2}, \quad f'(x) = \frac{-2x}{(1+x^2)^2}, \quad f''(x) = -2 \frac{1-3x^2}{(1+x^2)^3}$$

и на отрезке $0 \leq x \leq 1$

$$|f''(x)| \leq \left| -2 \frac{1-3x^2}{(1+x^2)^3} \right| \leq 2.$$

По условию погрешность не должна превышать 0,005. Отсюда в силу (23)

имеем
$$\frac{1}{12n^2} \cdot 2 \leq 0,005$$

или
$$n^2 \geq \frac{200}{6}.$$

Для удобства вычислений возьмем $n = 10$, тогда шаг $h = 0,1$. Вычислим значения y_i с 3 десятичными знаками после запятой. Результаты вычислений запишем в виде таблицы (табл. 3).

Таблица 3

i	x_i	$y_i = \frac{1}{1+x^2}$
0	0	1,000
1	0,1	0,990
2	0,2	0,962
3	0,3	0,917
4	0,4	0,862
5	0,5	0,800
6	0,6	0,735
7	0,7	0,671
8	0,8	0,610
9	0,9	0,552
10	1	0,500

Отсюда в силу формулы (24) получим

$$\int_0^1 \frac{dx}{1+x^2} \approx 0,1 \cdot (0,5 \cdot (1+0,5) + 7,099) \approx 0,785.$$

9.4. ЗАДАНИЯ К РАБОТЕ

Вычислить значение определенного интеграла с заданной точностью, используя метод Симпсона.

$$\int_a^b f(x) dx$$

Варианты заданий

№ варианта	Подынтегральная функция	Отрезок интегрирования		Заданная точность
		a	b	
1	$\cos(x + x^3)$	0	1	10^{-4}
2	$\sin(x^4 + 2x^3 + x^2)$	0,2	1,5	10^{-4}
3	$\exp(\sin(x))$	0	1	10^{-4}
4	$\sin(x) \cdot \exp(-x^2)$	0	2	10^{-4}
5	$\exp(\cos(x))$	0	1	10^{-4}
6	$\sin(x^2)$	0	1	10^{-4}
7	$\cos(x^2)$	0	1	10^{-4}
8	$\sin(x + x^3)$	0	1	10^{-4}
9	$\cos(x) \cdot \exp(-x^2)$	0	1	10^{-4}
10	$\sin(2x) \cdot \exp(-x^2)$	1	2	10^{-4}
11	$\exp(-(x + 1/x))$	1	2	10^{-4}
12	$\ln(x)/(1+x)$	1	2	10^{-4}
13	$\sqrt{x} \exp(-x^2)$	$\pi/2$	π	10^{-4}
14	$\cos(x^3)$	0	1	10^{-4}
15	$\ln(\sin(x))$	$\pi/4$	$\pi/2$	10^{-4}
16	$\cos(2 \sin(x))$	0	π	10^{-4}
17	$x^2 \exp(-x^2)$	0	π	10^{-4}
18	$x^4 \exp(-x^2)$	0	π	10^{-4}
19	$\sin(x^3)$	1	2	10^{-4}

20	$\ln(1+x)/x$	1	2	10^{-4}
21	$\exp(x)/x$	1	2	10^{-4}
22	$\exp(x^2)$	1	2	10^{-4}
23	$x \sin(x^3)$	0	$\pi/4$	10^{-4}
24	$\ln(1 + \cos(x))$	0	$\pi/4$	10^{-4}
25	$x \cos(x^3)$	0	$\pi/4$	10^{-4}
26	$\cos(2x + x^2)$	0	1	10^{-4}
27	$\sin(2x^4 + x^2 + x)$	0,2	1,5	10^{-4}
28	$\exp(\operatorname{tg}(x))$	0	1	10^{-4}
29	$\sin(-x) \cdot \exp(x^2)$	0	2	10^{-4}
30	$\exp(\cos(x))$	0	1	10^{-4}

9.5. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теорией приближенных вычислений определенных интегралов и разобрать приведенные примеры.
2. Выполнить ручной расчет заданного интеграла методом трапеций, как показано в примере 7.
3. Ознакомиться со схемой алгоритма метода решения задачи (рис. 17).
4. Пользуясь программой 6 в приложении 1, провести на компьютере расчет заданного интеграла методом Симпсона (начальное число разбиений $n=1$).
5. Сравнить результаты ручного и компьютерного расчета.

9.6. ОТЧЕТНОСТЬ

Работа засчитывается после представления студентом аккуратно оформленного отчета и при умении применять на практике методы приближенного вычисления интегралов.

Отчет по лабораторной работе должен содержать:

- Постановку задачи.
- Необходимый теоретический материал.
- Ручной расчет заданного интеграла методом трапеций.
- Схему алгоритма решения задачи методом Симпсона.
- Текст программы на Паскале.
- Результаты расчета на компьютере.
- Анализ полученных результатов.

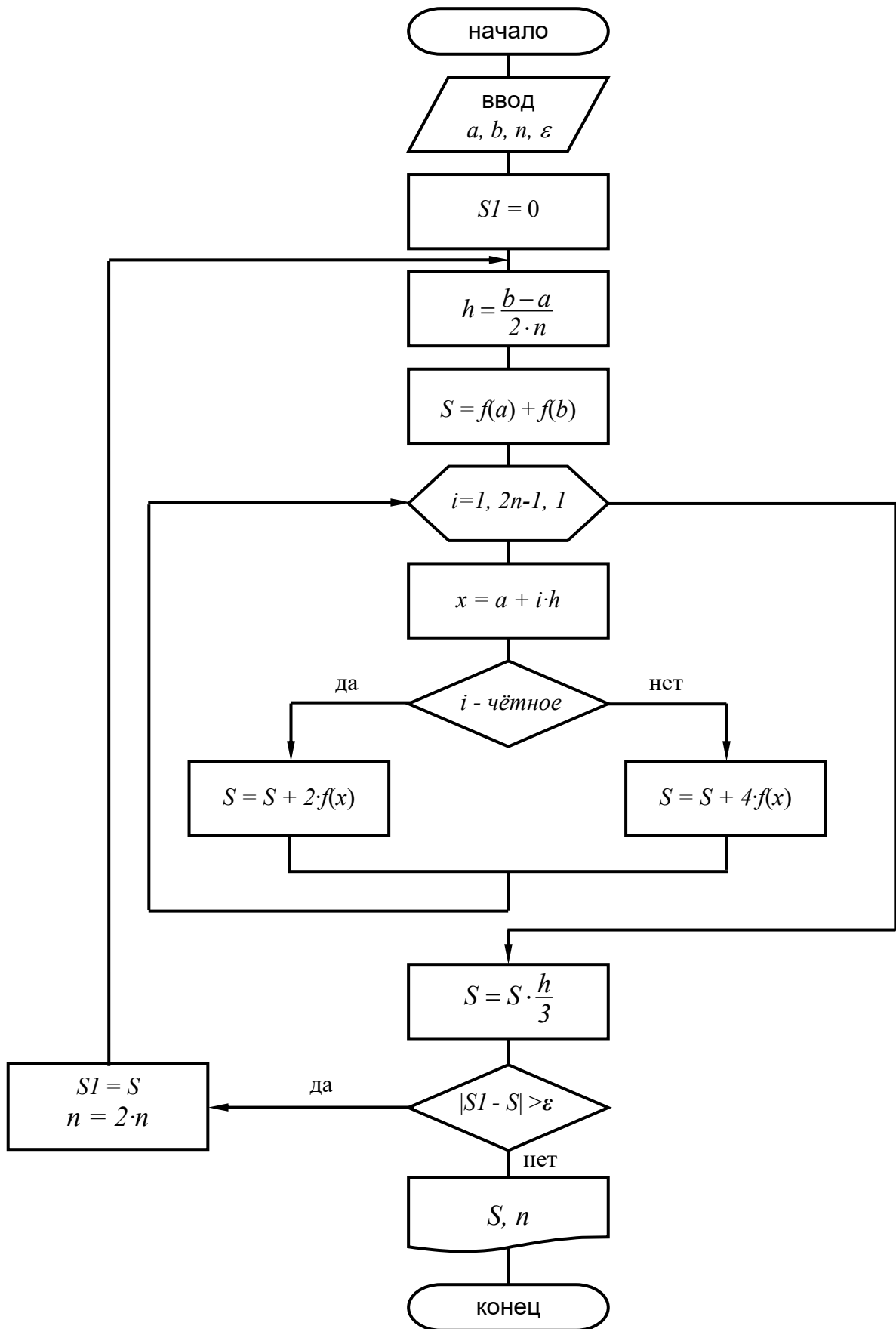


Рис. 17. Алгоритм вычисления интеграла по формуле Симпсона с заданной точностью

9.7. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается метод прямоугольников приближенного вычисления определенных интегралов?
2. В чем заключается метод трапеций приближенного вычисления интегралов?
3. Расскажите сущность метода параболических трапеций (Симпсона).
4. Как достигается необходимая точность при вычислении интеграла?
5. Какие разновидности метода прямоугольников вы знаете?

10. РАБОТА №10. ПРИБЛИЖЕННЫЕ МЕТОДЫ ВЫЧИСЛЕНИЯ КРАТНЫХ ИНТЕГРАЛОВ

10.1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение студентами умения приближенного вычисления кратных интегралов.

10.2. КРАТКИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

10.2.1. Общие сведения

Проблема вычисления кратных интегралов существенно отличается от случая простых интегралов. Если в однократном интеграле область интегрирования простая, ею является отрезок, то для кратных интегралов область интегрирования может быть достаточно сложной. В частности, для двукратного интеграла эта область геометрически представляет собой произвольную плоскую фигуру, а значение интеграла - есть объем трехмерной фигуры (рис. 18).

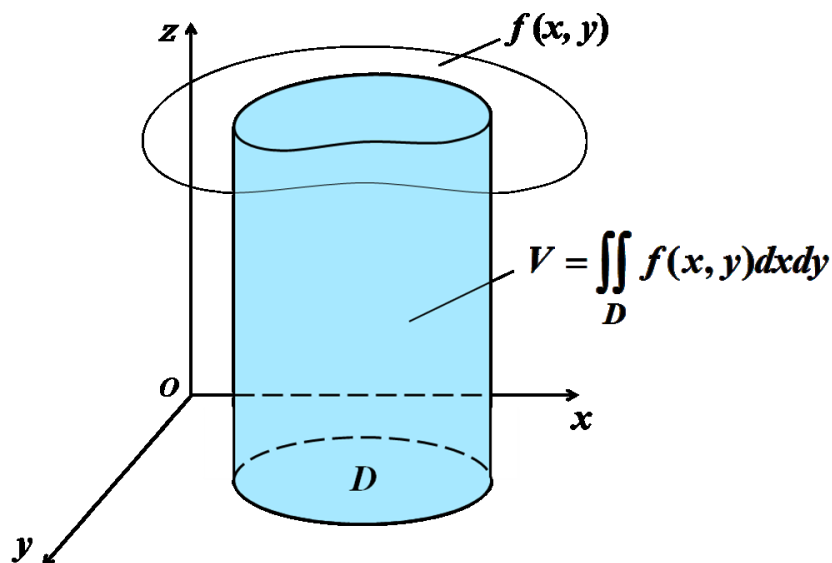


Рис. 18. Геометрический смысл двукратного интеграла: D – область интегрирования

Рассмотрим наиболее часто встречающиеся приближенные методы решения кратных интегралов на примере вычисления двукратного интеграла. Выбор того или иного метода определяется видом области интегрирования.

10.2.2. Метод ячеек

Данный метод применяется для вычисления двукратного интеграла, когда область интегрирования D представляет собой прямоугольник:

$$\iint_D f(x,y) dx dy, \quad D: \begin{cases} a \leq x \leq b \\ c \leq y \leq d \end{cases}. \quad (26)$$

Геометрически задача сводится к вычислению объема криволинейного параллелепипеда (рис. 19).

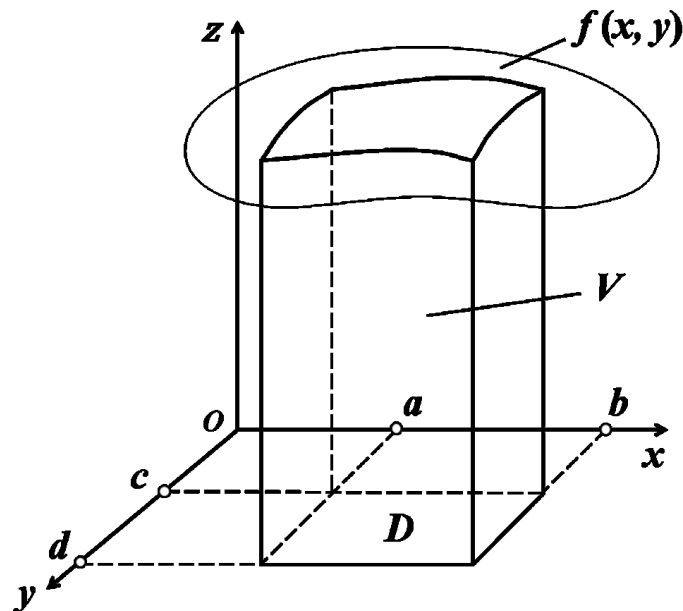


Рис. 19. Двукратный интеграл по прямоугольной области D

Идея вычисления интеграла (26) состоит в следующем. Поскольку в геометрии нет готовой формулы для вычисления объема произвольного криволинейного параллелепипеда, то эту фигуру нужно приближенно заменить на прямоугольный параллелепипед, объем которого легко вычисляется. Для снижения погрешности исходную фигуру сначала разбивают на множество более мелких, вычисляют каждый элементарный объем и складывают. Метод иллюстрирует рисунок 20. Область интегрирования D разбивают на ячейки (n – число разбиений по оси Ox , m – по оси Oy). Соответственно исходный объем разбивается на $n \times m$ элементарных объемов.

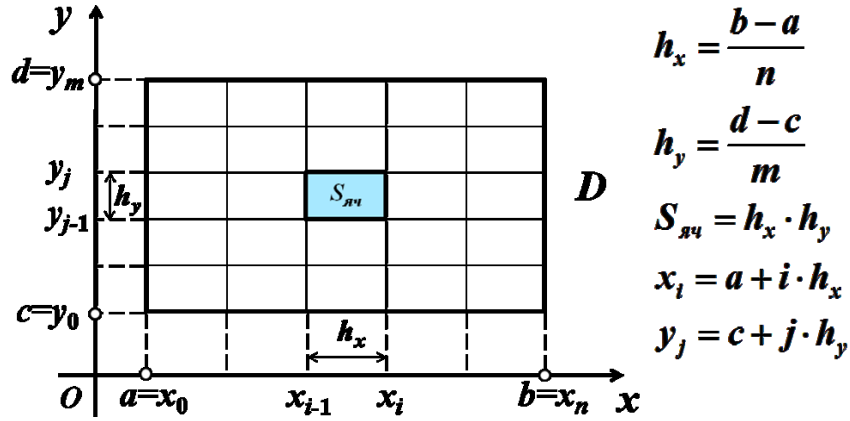


Рис. 20. Разбиение области D на ячейки

Для вычисления каждого элементарного объема площадь ячейки $S_{яч}$ умножают на значение подынтегральной функции $f(\bar{x}_i, \bar{y}_j)$ в центре каждой ячейки:

$$\bar{x}_i = \frac{x_i + x_{i-1}}{2}, \quad i = 1, \dots, n; \quad \bar{y}_j = \frac{y_j + y_{j-1}}{2}, \quad j = 1, \dots, m.$$

Суммируя элементарные объемы, получают приближенное значение двукратного интеграла

$$\begin{aligned} \iint_D f(x, y) dx dy = V &\approx S_{яч} \cdot f(\bar{x}_1, \bar{y}_1) + \dots + S_{яч} \cdot f(\bar{x}_n, \bar{y}_m) = \\ &= h_x h_y \sum_{i=1}^n \sum_{j=1}^m f(\bar{x}_i, \bar{y}_j) \end{aligned} \quad (27)$$

Формула (27) называется формулой ячеек. Алгоритм вычисления интеграла по методу ячеек приведен на рис. 21.

Формула ячеек имеет второй порядок точности $O(n^{-2} + m^{-2})$ (ошибка обратно пропорциональна числу разбиений во второй степени). Для оценки погрешности вычислений можно применить метод Рунге. Метод Рунге состоит в следующем. Приблизленно вычисляем интеграл по формуле (27) при заданном числе разбиений n и m . Полученное значение обозначим $I_{(n,m)}$. Затем снова вычисляем интеграл по этой же формуле, но при удвоенном числе разбиений $2n$ и $2m$. Обозначим значение $I_{(2n,2m)}$. Погрешность Δ полученного значения принимают равной

$$\Delta = \frac{|I_{(2n,2m)} - I_{(n,m)}|}{15}. \quad (28)$$

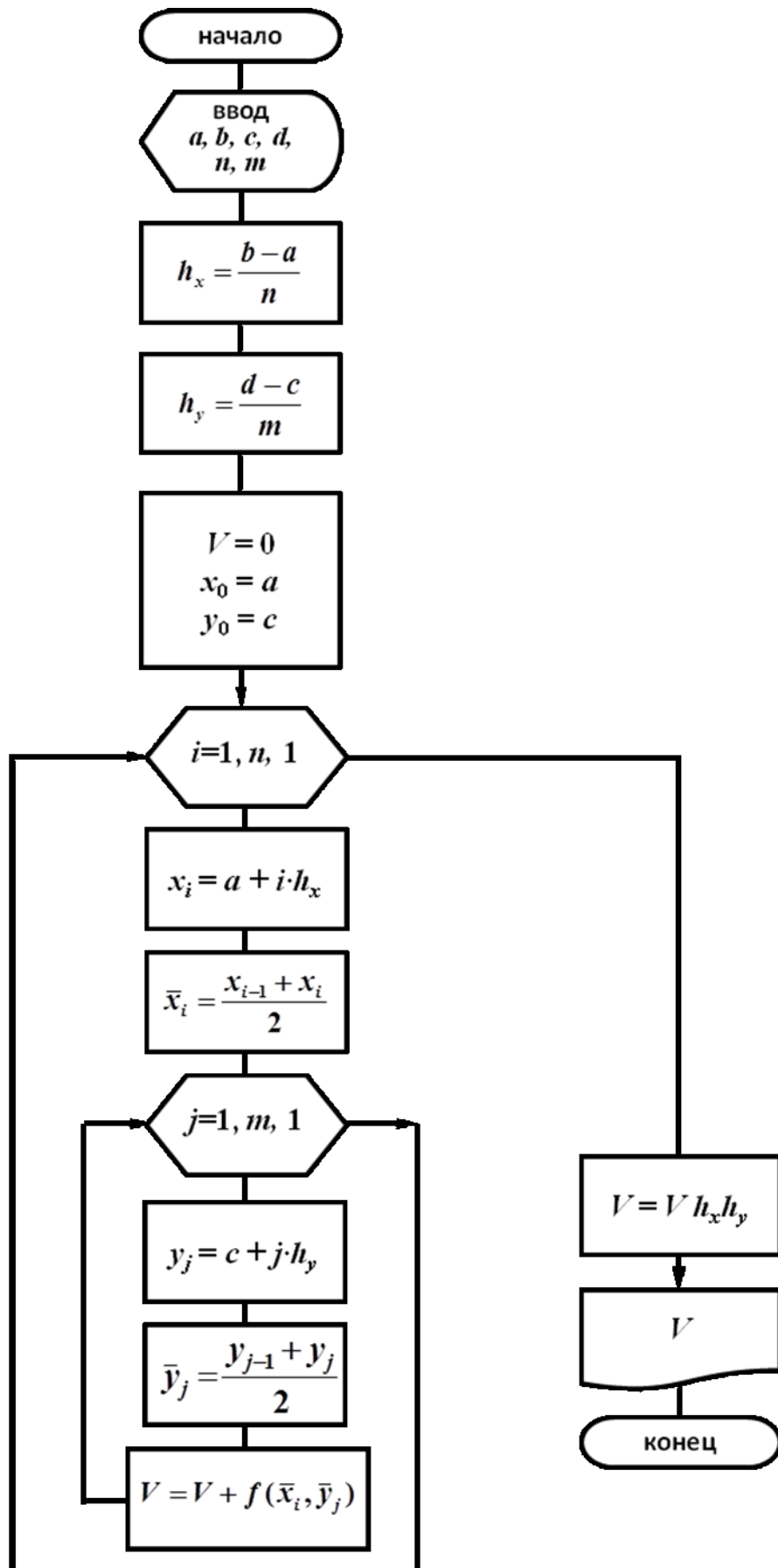


Рис. 21. Схема алгоритма метода ячеек

10.2.3. Метод последовательного интегрирования

В отличие от метода ячеек метод последовательного интегрирования применяется, когда область интегрирования D имеет сложную форму. Идея вычисления заимствована из классической математики.

Пусть для двукратного интеграла область интегрирования D ограничена непрерывными кривыми $y=\varphi(x)$, $y=\psi(x)$, $\varphi(x)\leq\psi(x)$ и двумя прямыми $x=a$, $x=b$ (рис. 22):

$$\iint_D f(x,y)dx dy, \quad D: \begin{cases} a \leq x \leq b \\ \varphi(x) \leq y \leq \psi(x) \end{cases}.$$

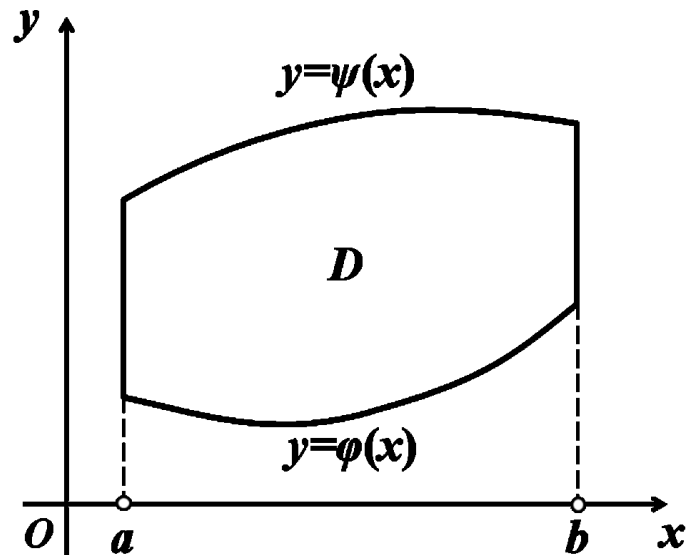


Рис. 22. Вид области интегрирования

В математике известно правило вычисления двукратного интеграла

$$I = \iint_D f(x,y)dx dy = \int_a^b dx \int_{\varphi(x)}^{\psi(x)} f(x,y)dy. \quad (29)$$

Обозначим внутренний интеграл как функцию

$$F(x) = \int_{\varphi(x)}^{\psi(x)} f(x,y)dy. \quad (30)$$

Тогда двукратный интеграл

$$I = \int_a^b F(x)dx. \quad (31)$$

Интегралы (30) и (31) являются однократными и для их вычисления можно применить соответствующие приближенные формулы (прямоугольников, трапеций или Симпсона).

Проведем разбиение области интегрирования D по оси Ox с шагом h_x . Каждая хорда (например, cd) есть проекция сечения искомого объема (рис. 23).

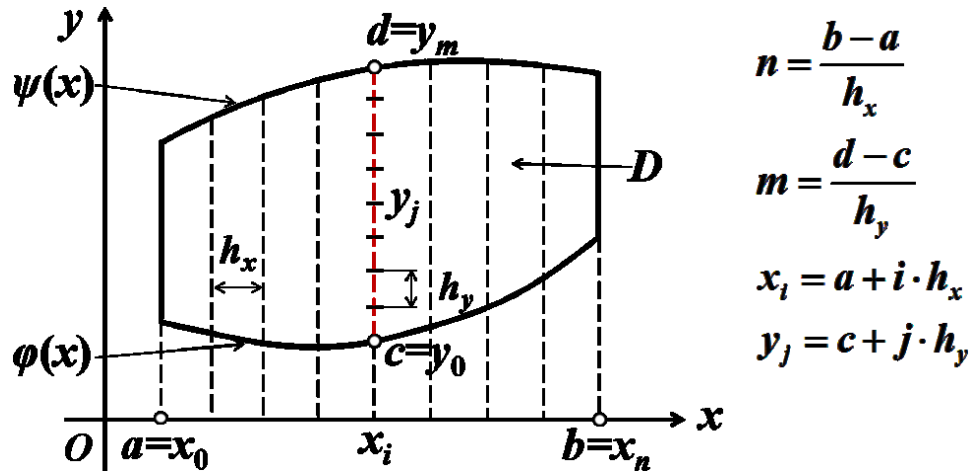


Рис. 23. Разбиение области интегрирования

Для приближенного вычисления внешнего интеграла (31) применим формулу трапеций

$$I = \iint_D f(x, y) dx dy = \int_a^b F(x) dx \approx h_x \left[\frac{F(a) + F(b)}{2} + \sum_{i=1}^{n-1} F(x_i) \right], \quad (32)$$

где $F(x_i)$ есть площади сечений исходной фигуры. Эти площади (значения внутреннего интеграла (30) в точках x_i) приближенно вычислим, также используя формулу прямоугольных трапеций. Для этого каждое сечение (в проекции - хорду) разбиваем вдоль оси Oy с шагом h_y .

$$F(x_i) = \int_{c=\varphi(x_i)}^{d=\psi(x_i)} f(x_i, y) dy \approx h_y \left[\frac{f(x_i, c) + f(x_i, d)}{2} + \sum_{j=1}^{m-1} f(x_i, y_j) \right], i = 0, 1, \dots, n. \quad (33)$$

Порядок расчета

Отрезок $[a, b]$ разбивают с шагом h_x , в каждой точке x_i вычисляют значения $c=\varphi(x_i)$, $d=\psi(x_i)$. Хорду $[c, d]$ разбивают с шагом h_y . Используя формулу (33) вычисляют площади сечений $F(x_i)$, $i=0, \dots, n$. По формуле (32) вычисляют значение двукратного интеграла.

Алгоритм вычисления интеграла по методу ячеек приведен на рис. 24.

Пример 8

Пользуясь методом последовательного интегрирования вычислить двукратный интеграл

$$\iint_D (1 - x^2 + \sqrt{1 + y^2}) dx dy, \quad D: x^2 + y^2 \leq 1.$$

Область интегрирования D ограничена окружностью $x^2 + y^2 = 1$. Учитывая симметрию подынтегральной функции и области интегрирования относительно начала координат, можно предложить следующую формулу вычисления искомого интеграла

$$\iint_{x^2+y^2 \leq 1} (1-x^2 + \sqrt{1+y^2}) dx dy = 4 \cdot \int_0^1 dx \int_0^{\sqrt{1-x^2}} (1-x^2 + \sqrt{1+y^2}) dy.$$

Из данной формулы видно, что достаточно вычислить четверть всего объема (четверть значения интеграла) для D : $0 \leq x \leq 1$, $\varphi(x) \leq y \leq \sqrt{1-x^2}$. Вычисления проводятся на компьютере по программе 8 (приложение 1) в соответствии с алгоритмом метода. Полученный результат следует умножить на четыре.

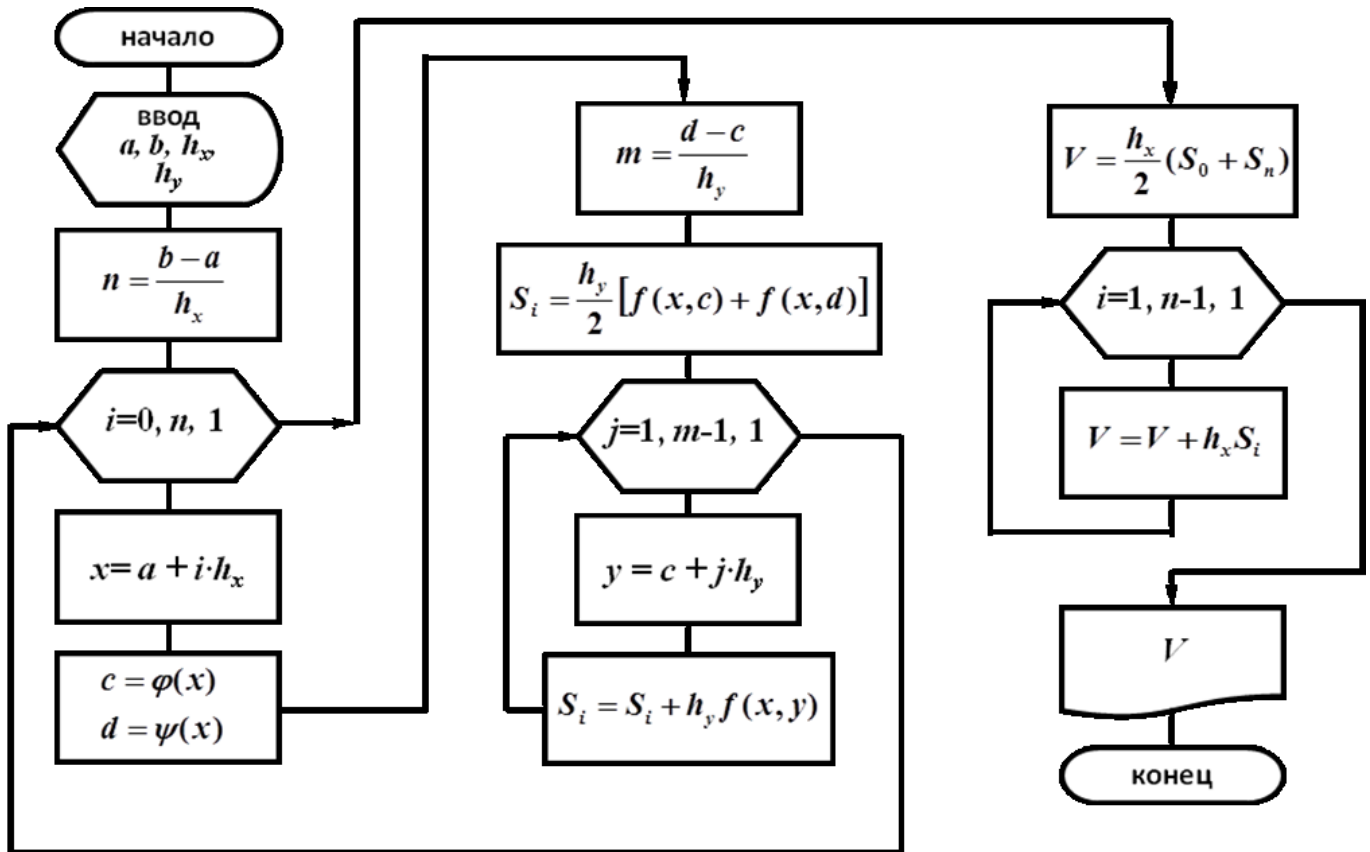


Рис. 24. Схема алгоритма метода последовательного интегрирования

10.3. ЗАДАНИЯ К РАБОТЕ

ЗАДАНИЕ 1

Методом ячеек вычислить приближенное значение двукратного интеграла

$$\iint_D f(x, y) dx dy, \quad D: \begin{cases} a \leq x \leq b \\ c \leq y \leq d \end{cases}$$

Варианты заданий

№ варианта	Подынтегральная функция $f(x,y)$	Границы области интегрирования D			
		a	b	c	d
1	$\cos(xy + x^3)$	0	1	1	2
2	$\sin(y^4 + 2yx^3 + x^2)$	0	1	1	2
3	$e^{\sin(xy)}$	0	1	0	1
4	$e^{\cos(x)+\sin(y)}$	0	1	0	1
5	$\sin(y) \cdot e^{-x^2}$	0	1	0	1
6	$\sin(x^2 + y^2)$	0	1	0	1
7	$\cos(x^2 + y^2)$	0	1	0	1
8	$\sin(xy + x^3)$	0	1	1	2
9	$\cos(x) \cdot e^{-y^2}$	0	1	0	1
10	$\sin(2xy) \cdot e^{-x^2y}$	0	1	0	1
11	$\sin(x + y) \cdot e^{-x/y}$	0	1	1	2
12	$\sqrt{x+y} \cdot e^{x^2+y^2}$	0	1	0	1
13	$x^2 + (y-1)^2$	0	1	0	1
14	$x^2 - (y-1)^2$	0	1	0	1
15	$(x-y+1)^2$	0	1	0	1
16	$\ln(x^2 + y^2)$	0	1	1	2
17	$\frac{x^2}{y} + \frac{y^2}{x}$	1	2	1	2
18	$1 - \sqrt{x^2 + y^2}$	0	1	0	1
19	$e^{-x} + e^{-y}$	0	1	0	1
20	$(e^{-x} - e^{-y})^2$	0	1	0	1
21	e^{-x}/y	0	1	1	2
22	e^{-xy}/xy	1	2	1	2
23	$\frac{1}{1+x^2+y^2}$	0	2	0	2

24	$e^{\cos(x)-\sin(y)}$	0	1	0	1
25	$\sin(x^2 - y^2)$	0	1	0	1
26	$\cos(x^2 - y^2)$	0	1	0	1
27	$\sin(xy - x^3)$	0	1	0	1
28	$\sin(xy - y^3)$	0	1	1	2
29	$\ln(x^2 - y^2)$	2	3	0	1
30	$e^{\frac{x+y}{xy}}$	1	2	1	2

ЗАДАНИЕ 2

Методом последовательного интегрирования вычислить приближенное значение двукратного интеграла

$$\iint_D f(x, y) dx dy, \quad D: \begin{cases} a \leq x \leq b \\ \varphi(x) \leq y \leq \psi(x) \end{cases}$$

Варианты заданий

№ вариан- та	Подынтегральная функция $f(x, y)$	Область интегрирования D
1	$\frac{1}{\sqrt{24 + x^2 + y^2}}$	$x^2 + y^2 \leq 25$
2	$\sqrt{1 + x^2 + y^2}$	$x^2 + y^2 \leq 9$
3	$\ln(1 + x^2 + y^2)$	$ x + y \leq 1$
4	$\arctg(x + y)$	$\begin{cases} 0 \leq x \leq 1 \\ -1 \leq y \leq 1 - x \end{cases}$
5	$\sqrt{4 - x^2 + y^2}$	$x^2 + y^2 \leq 4$
6	$\frac{1}{1 + \cos^2(x) + \sin^2(y)}$	$ x + y \leq 1$
7	$x^2 + (y - 1)^2$	$\begin{cases} 0 \leq x \leq 2 \\ x \leq y \leq 2x \end{cases}$
8	$x^2 - (y - 1)^2$	$\begin{cases} -2 \leq x \leq 2 \\ x^2 - 1 \leq y \leq 2x \end{cases}$

9	$(x - y + 1)^2$	$\begin{cases} -1 \leq x \leq 1 \\ -\sqrt{1-x^2} \leq y \leq 1-x^2 \end{cases}$
10	$x^4 - x^2 + y^4$	$\begin{cases} 0 \leq x \leq 1 \\ x^3 \leq y \leq x \end{cases}$
11	$x^2 - 2xy - y^2$	$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq 1-x \end{cases}$
12	$xy + \frac{5}{x} + \frac{2}{y}$	$\begin{cases} -1 \leq x \leq 2 \\ 2-x \leq y \leq \sqrt{2x-x^2} \end{cases}$
13	$1 - \sqrt{x^2 + y^2}$	$\begin{cases} 1 \leq x \leq 10 \\ 0 \leq y \leq \lg(x) \end{cases}$
14	$e^{2x+3y}(x^2 + y^2)$	$\begin{cases} 0 \leq x \leq 1 \\ x \leq y \leq 2x \end{cases}$
15	$e^{-(x^2+xy+y^2)}$	$x^2 + y^2 \leq 4$
16	$\sin(x) + \cos(y) + x^2 y^2$	$\begin{cases} 0 \leq x \leq 2\pi \\ 0 \leq y \leq \sin(x) \end{cases}$
17	$\sin(x)\sin(y)\sin(x+y)$	$\begin{cases} 0 \leq x \leq 1 \\ x^2 \leq y \leq x \end{cases}$
18	$x - 2y + \ln(x^2 + y^2)$	$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq \sin(x) \end{cases}$
19	$xy \ln(\sqrt{x+y})$	$1 \leq x^2 + y^2 \leq 4$
20	$x + y + 4\sin(x)\sin(y)$	$\begin{cases} -1 \leq x \leq 1 \\ x^2 \leq y \leq 1 \end{cases}$
21	$(x^2 + y^2)e^{-(x^2+y^2)}$	$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq x^2 \end{cases}$
22	$x^2 + y^2 + 2x + 4y$	$\begin{cases} 0 \leq x \leq 1 \\ 0 \leq y \leq x^2 \end{cases}$
23	$x + \frac{y^2}{4x+1}$	$\left\{ \frac{x^2}{4} + \frac{y^2}{3} \leq 1 \right\}$
24	$xy^2(x+2y)$	$\begin{cases} x \leq 1 \\ 0 \leq y \leq x^2 \end{cases}$

25	$\frac{x^2}{y} + \frac{y^2}{x+1}$	$\begin{cases} 0 \leq x \leq 1 \\ -1-x \leq y \leq \sqrt{2-x^2} \end{cases}$
26	$\frac{x+y}{\sqrt{2}} - x^2 - y^2$	$\begin{cases} 0 \leq x \leq 2 \\ x \leq y \leq 2x \end{cases}$
27	$\frac{e^{-x^2} - e^{-y^2}}{x}$	$\begin{cases} 0 \leq x \leq e \\ 0 \leq y \leq \ln(x) \end{cases}$
28	$\frac{(e^{-x} - e^{-y})^2}{x}$	$\begin{cases} 1 \leq x \leq 2 \\ 0 \leq y \leq \sqrt{x^2 - 1} \end{cases}$
29	$\frac{e^{-x} - e^{-y}}{xy}$	$1 \leq x^2 + y^2 \leq 4$
30	$\frac{\ln(1-x^2)}{\sqrt{1-xy}}$	$x^2 + y^2 \leq 0,25$

10.4. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Ознакомиться с теорией приближенных вычислений кратных интегралов.
2. Пользуясь приведенной в приложении 1 программой 7, выполнить на компьютере расчет интеграла из задания 1 методом ячеек при числе разбиений $n=10$ и $m=10$.
3. Выполнить повторный расчет этого интеграла тем же методом при числе разбиений $n=20$ и $m=20$.
4. Пользуясь методом Рунге по формуле (28) оценить погрешность Δ вычисленного значения интеграла.
5. Для интеграла из задания 2 графически изобразить область интегрирования D и определить функции границ области.
6. Пользуясь приведенной в приложении 1 программой 8, выполнить на компьютере расчет интеграла из задания 2 методом последовательного интегрирования при шаге разбиений $h_x=0,1$ и $h_y=0,1$.

10.5. ОТЧЕТНОСТЬ

Работа засчитывается после представления студентом аккуратно оформленного отчета и при умении применять на практике методы приближенного вычисления интегралов.

Отчет по лабораторной работе должен содержать:

- Постановку задачи из задания 1.
- Необходимый теоретический материал.
- Схему алгоритма решения задачи методом ячеек.

- Текст программы на Паскале.
- Результаты расчета интеграла на компьютере при разном числе разбиений.
- Расчет погрешности значения интеграла по формуле Рунге.
- Постановку задачи из задания 2.
- Необходимый теоретический материал.
- Рисунок области интегрирования.
- Схему алгоритма решения задачи методом последовательного интегрирования.
- Текст программы на Паскале.
- Результат вычисления интеграла на компьютере.

10.6. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается метод ячеек приближенного вычисления двукратных интегралов?
2. Применение метода ячеек для вычисления интегралов большей кратности.
3. Расскажите сущность метода последовательного интегрирования.
4. Как достигается необходимая точность приближенного вычисления интеграла?
5. В чем заключается метод Рунге для оценки погрешности вычисления?

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Информатика: базовый курс / ред. С. В. Симонович. - 2-е изд. - СПб. : Питер, 2010. - 640 с.
2. Методы вычислительной математики для решения инженерных задач в нефтегазовой отрасли: учеб. пособие / А. К. Ращепкин, В. Б. Салагаев. - Уфа : Изд-во УГНТУ, 2005. - 99 с.
3. Зеленьяк, О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения. учеб. пособие / О.П. Зеленьяк. –М.: ДМК Пресс, 2009. -312с.
4. Канцедаль, С. А. Алгоритмизация и программирование: учеб. пособие / С.А. Канцедаль. - М. : ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 352 с.
5. Голицина, О.Л. Языки программирования: учеб. пособие / О.Л. Голицина, Т.Л. Партыка, И.И. Попов. - М.: ФОРУМ, 2010. - 400 с.
6. Ляхович, В. Ф. Основы информатики: учебник / В. Ф. Ляхович, В.А. Молодцов, Н.Б. Рыжикова. -М.: КНОРУС, 2016. -348 с.

Приложение 1

Программа 1. Уточнение корня уравнения методом половинного деления

```

program mpd;
var a,b,x,eps:real;
function f(x:real):real;
begin
  f:=exp(x)+x;      {здесь записывается функция f(x) уравнения f(x)=0}
end;
begin
  writeln('интервал поиска корня:');
  write('a='); readln(a);
  write('b='); readln(b);
  write('погрешность вычисления eps='); readln(eps);
  repeat
    x:=(a+b)/2;      {деление отрезка}
    if f(a)*f(x)<0 then b:=x else a:=x; {перенос границы в точку деления}
  until (b-a)<=eps;
  x:=(a+b)/2;
  writeln('результат уточнения корня x=',x:10:5);
end.

```

Программа 2. Уточнение корня уравнения методом хорд и касательных

```

program mhk;
var a,b,x,eps:real;
function fnx(x:real):real;
begin
  fnx:=sqr(x)-2;      {здесь записывается функция f(x) уравнения f(x)=0}
end;
function fnp(x:real):real;
begin
  fnp:=2*x;           {здесь записывается первая производная функции f(x)}
end;
begin
  writeln('интервал поиска корня:');
  write('a='); readln(a); write('b='); readln(b);
  write('погрешность вычисления eps='); readln(eps);
  repeat
    a:=a-fnx(a)*(b-a)/(fnx(b)-fnx(a)); {уравнение хорды}
    b:=b-fnx(b)/fnp(b);                {уравнение касательной}
  until abs((b-a))<=eps;
  x:=(a+b)/2;
  writeln('результат уточнения корня x=',x:10:5);
end.

```

Программа 3. Уточнение корня уравнения методом простой итерации

```

program mpi;
uses crt;
var x,y,eps:real;
label 1;           {объявление метки}
function g(x:real):real;
begin
  g:=sqr(x)/3;      {здесь записывается функция g(x) уравнения x=g(x)}
end;
begin
  clrscr;
  writeln('начальное приближение к корню:');
  write('x='); readln(x);
  write('погрешность вычисления eps='); readln(eps);
  1: y:=g(x);
     if abs(y-x)>eps then begin x:=y; goto 1 end;
  writeln('результат уточнения корня x=',x:10:5);
end.

```

Программа 4. Решение системы нелинейных уравнений методом итераций

```

program mpi_sys;
var x1,xn1,x2,xn2,eps:real; k:integer;
label 1;
begin
  writeln('начальные значения для поиска решения:');
  write('x1='); readln(x1); write('x2='); readln(x2);
  write('погрешность вычисления eps='); readln(eps);
  k:=1;
  1: xn1 := 1/3*cos(x2)+0.3;      {первое уравнение системы}
     xn2 := sin(x1-0.6)-1.6;     {второе уравнение системы}
     if (abs(xn1-x1)>eps) or (abs(xn2-x2)>eps) then
       begin
         x1:=xn1; x2:=xn2; k:=k+1; goto 1;
       end;
  writeln('решение системы нелинейных уравнений');
  writeln('x1=',x1:10:5,' x2=',x2:10:5);
  writeln('число итераций k=',k:4);
end.

```

Программа 5. Решение системы линейных уравнений методом Гаусса

```

program gauss;
var s:real; i,j,k,n:integer;
x:array[1..10] of real;
a:array[1..10,1..11] of real;

```

```

begin
write('число уравнений n='); readln(n);
writeln('значения коэффициентов системы:');
for i:=1 to n do
for j:=1 to n+1 do
begin
write('a(',i:2,',',j:2,')='); readln(a[i,j]);
end;
for k:=1 to n-1 do {прямой ход метода}
begin
for j:=k+1 to n+1 do
a[k,j]:=a[k,j]/a[k,k];
a[k,k]:=1;
for i:=k+1 to n do
begin
for j:=k+1 to n+1 do
a[i,j]:=a[i,j]-a[k,j]*a[i,k];
a[i,k]:=0;
end;
end;
if abs(a[n,n])>1e-50 then
begin
x[n]:=a[n,n+1]/a[n,n]; {обратный ход метода}
for i:=n-1 downto 1 do
begin
s:=0;
for j:=i+1 to n do s:=s+a[i,j]*x[j];
x[i]:=a[i,n+1]-s;
end;
writeln('решение системы:');
for i:=1 to n do
begin writeln('x(',i:2,')=',x[i]:10:5); end;
end
else
writeln('система решения не имеет');
end.

```

Программа 6. Приближенное вычисление интеграла методом Симпсона

```

program simpson;
var a,b,s,s1,eps:real; n:integer;
label 1;
function integral(n:integer):real; {вычисление интеграла по формуле Симпсона}
var h,x:real; i:integer;
function f(x:real):real;
begin
f:=1/(1+sqr(x)); {подынтегральная функция}

```

```

end;
begin
h:=(b-a)/(2*n);
s:=f(a)+f(b);
for i:=1 to 2*n-1 do
  begin
    x:=a+i*h;
    if i/2=int(i/2) then s:=s+2*f(x)
    else s:=s+4*f(x);
  end;
integral:=s*h/3;
end;
BEGIN    {начало основной программы}
writeln('исходные данные:');
write('нижний предел a='); readln(a);
write('верхний предел b='); readln(b);
write('начальное число разбиений n='); readln(n);
write('требуемая погрешность вычисления eps='); readln(eps);
s1:=0;
1: s:=integral(n);
if abs(s-s1)>eps then begin n:=n*2; s1:=s; goto 1; end;
writeln('значение интеграла s=',s:2:5);
writeln('при числе разбиений n=',n:6);
END.

```

Программа 7. Приближенное вычисление интеграла методом ячеек

```

program cells;
var a,b,c,d,hx,hy,xc,yc,v:real;
x,y:array[0..2000] of real;
i,j,n,m:integer;
function f(x,y:real):real;
begin
f:=2*sqr(x)+sqr(y);           {подынтегральная функция}
end;
BEGIN
writeln('Границы области интегрирования:');
write('a='); readln(a);
write('b='); readln(b);
write('c='); readln(c);
write('d='); readln(d);
writeln('Число разбиений по осям:');
write('n='); readln(n);
write('m='); readln(m);
hx:=(b-a)/n; hy:=(d-c)/m;
x[0]:=a; y[0]:=c;
v:=0;

```

```

for i:=1 to n do
  begin
    x[i]:=a+i*hx;
    xc:=(x[i]+x[i-1])/2;
    for j:=1 to m do
      begin
        y[j]:=c+j*hy;
        yc:=(y[j]+y[j-1])/2;
        v:=v+f(xc,yc);
      end;
    end;
  v:=v*hx*hy;
  writeln('Значение интеграла методом ячеек v=',v:2:5);
END.

```

Программа 8. Приближенное вычисление интеграла методом последовательного интегрирования

```

program posint;
var a,b,c,d,x,y,hx,hy,v:real;
s:array[0..2000] of real; i,j,n,m:integer;
function f(x,y:real):real;
begin
  f:= 1-sqr(x)+sqrt(1+sqr(y));           {подынтегральная функция}
end;
function fi(x:real):real;
begin
  fi:=0;                                {функция нижней границы области интегрирования}
end;
function psi(x:real):real;
begin
  psi:= sqrt(1-sqr(x));                 {функция верхней границы области интегрирования}
end;
BEGIN
  writeln('Границы области интегрирования по оси X:');
  write('a='); readln(a); write('b='); readln(b);
  writeln('Шаг разбиений по осям:');
  write('hx='); readln(hx); write('hy='); readln(hy);
  n:=trunc((b-a)/hx);
  for i:=0 to n do
    begin
      x:=a+i*hx;
      c:=fi(x); d:=psi(x);
      m:=trunc((d-c)/hy);
      s[i]:=(f(x,c)+f(x,d))/2;
    end
  end

```

```

for j:=1 to m-1 do
  begin
    y:=c+j*hy;
    s[i]:=s[i]+f(x,y);
  end;
  s[i]:=s[i]*hy;
end;
v:=(s[0]+s[n])/2;
for i:=1 to n-1 do v:=v+s[i];
v:=v*hx;
writeln('значение интеграла методом послед. интегрирования v=',v:2:5);
END.

```

СОДЕРЖАНИЕ

Введение.....	1
Среда программирования «PascalABC.net».....	2
1. Работа №1. Программирование алгоритмов линейной структуры.....	3
2. Работа №2. Программирование алгоритмов разветвленной структуры.....	8
3. Работа №3. Программирование алгоритмов циклической структуры.....	15
4. Работа №4. Программирование итерационных циклов. Суммирование функциональных рядов.....	20
5. Работа №5. Программирование операций с одномерными и двумерными массивами.....	27
6. Работа №6. Численное решение нелинейного уравнения.....	36
7. Работа №7. Численное решение системы нелинейных уравнений.....	46
8. Работа №8. Решение системы линейных алгебраических уравнений.....	53
9. Работа №9. Приближенное вычисление простых интегралов.....	62
10. Работа №10. Приближенные методы вычисления кратных интегралов.....	71
Список рекомендуемой литературы.....	82
Приложение 1	83