# Оглавление

## Предисловие

Цель дисциплины ”Прикладная математика” - изучение методологии приложений фундаментальных научных достижений в математике и информатике к решению прикладных задач с учетом возрастающих возможностей компьютерной техники и современных компьютерных технологий.

Выполнение лабораторных работ направлено на формирование и проверку освоения компетенций, предусмотренных рабочей программой дисциплины:

**ПК-9** способностью получать и обрабатывать информацию из различных источников, используя современные информационные технологии и критически ее осмысливать

*(код и наименование)*

В результате изучения дисциплины (модуля) обучающийся должен:

***Знать:***

- о точных и приближенных методах решения;
- о связи задач дифференциального и интегрального исчисления;
- основные понятия и методы решения уравнений математической физики; возможные сферы их приложений;

***Уметь:***

- применять теорию дифференциальных уравнений в частных производных в решении научных задач;
- применять математические модели для типичных базовых задач и проводить необходимые расчеты в рамках построенных моделей.

***Владеть***:

- навыками применения математических методов в прикладных задачах;

навыками использования аналитического и приближенного решения дифференциальных уравнений в частных производных.

**ПК-12** способностью использовать современные достижения науки и передовых информационных технологий в научно-исследовательских работах

*(код и наименование)*

В результате изучения дисциплины (модуля) обучающийся должен:

***Знать:***

- современные тенденции развития, научные и прикладные достижения прикладной математики и информатики
- методологию построения математических моделей явлений и объектов, относящихся к профилю деятельности, с использованием аппарата прикладной математики

***Уметь:***

- строить математические модели явлений и объектов, относящихся к профилю деятельности

***Владеть:***

- математическим аппаратом для разработки математических моделей явлений и объектов, относящихся к профилю деятельности

Материал дисциплины предназначен для использования в курсах, связанных с постановкой и решением реальных задач (например, различные разделы теоретической и прикладной механики, расчет строительных конструкций), с построением математических моделей физических процессов, верификацией гипотез, теоретических моделей и т.д. Знания, полученные в процессе изучения дисциплины, могут быть использованы в курсах профильной направленности.

К **основным задачам**, дисциплины следует отнести:

– изучение основных понятий дисциплины;

– изучение современных численных методов;

– изучение возможностей различных численных алгоритмов, методов, особенностей их применения при решении прикладных задач;

– различные численные методы, их характеристики и свойства, особенности применения этих методов для решения практических задач.

**Объектом изучения** научной и учебной дисциплины " Прикладная математика " являются исследование практических задач в области строительства.

## Лабораторная работа №1

Тема: **Выполнения математических, инженерных и технических расчетов в системе компьютерной математики SciLab.**

1. **Цель работы**

Закрепление, углубление и совершенствование знаний и практических навыков работы на персональном компьютере с использованием современных компьютерных технологий.

2. **Учебные вопросы, подлежащие рассмотрению:**

– Основы работы в Scilab
– Функции в Scilab
– Массивы в Scilab
– Построение двумерных графиков

3. **Порядок выполнения работы**

**Основы работы в Scilab**

Scilab - это система компьютерной математики, которая предназначена для выполнения инженерных и научных вычислений, таких как:

- решение нелинейных уравнений и систем;
- решение задач линейной алгебры;
- решение задач оптимизации;
- дифференцирование и интегрирование;
- обработка экспериментальных данных (интерполяция и аппроксимация, метод наименьших квадратов);
- решение обыкновенных дифференциальных уравнений и систем.

Кроме того, Scilab предоставляет широкие возможности по созданию и редактированию различных видов графиков и поверхностей.

После запуска Scilab на экране появится основное окно приложения. Окно содержит меню, панель инструментов и рабочую область. Признаком того, что система готова к выполнению команды, является наличие знака приглашения > в командной строке, после которого расположен активный (мигающий) курсор.

Ввод команд в Scilab осуществляется с клавиатуры. Нажатие клавиши Enter заставляет систему выполнить команду и вывести результат (рис. 1).

Исправить что-либо в области просмотра уже выполненных команд нельзя. Однако все ранее вводимые команды сохраняются в специальной области памяти, и их можно просмотреть с помощью клавиш клавиатуры | |. Например, нажатие клавиши | один раз в пустой в командной строке отразит предыдущую выполненную команду, которую можно отредактировать и запустить заново.

Если набираемое выражение очень длинное, его можно продолжить на следующей строке, для этого в месте прерывания нужно набрать три точки без пробелов, а затем продолжить набор выражения на следующей строке.

Для подавления вывода на экран результатов промежуточных вычислений, в конце команды используется точку с запятой «;».

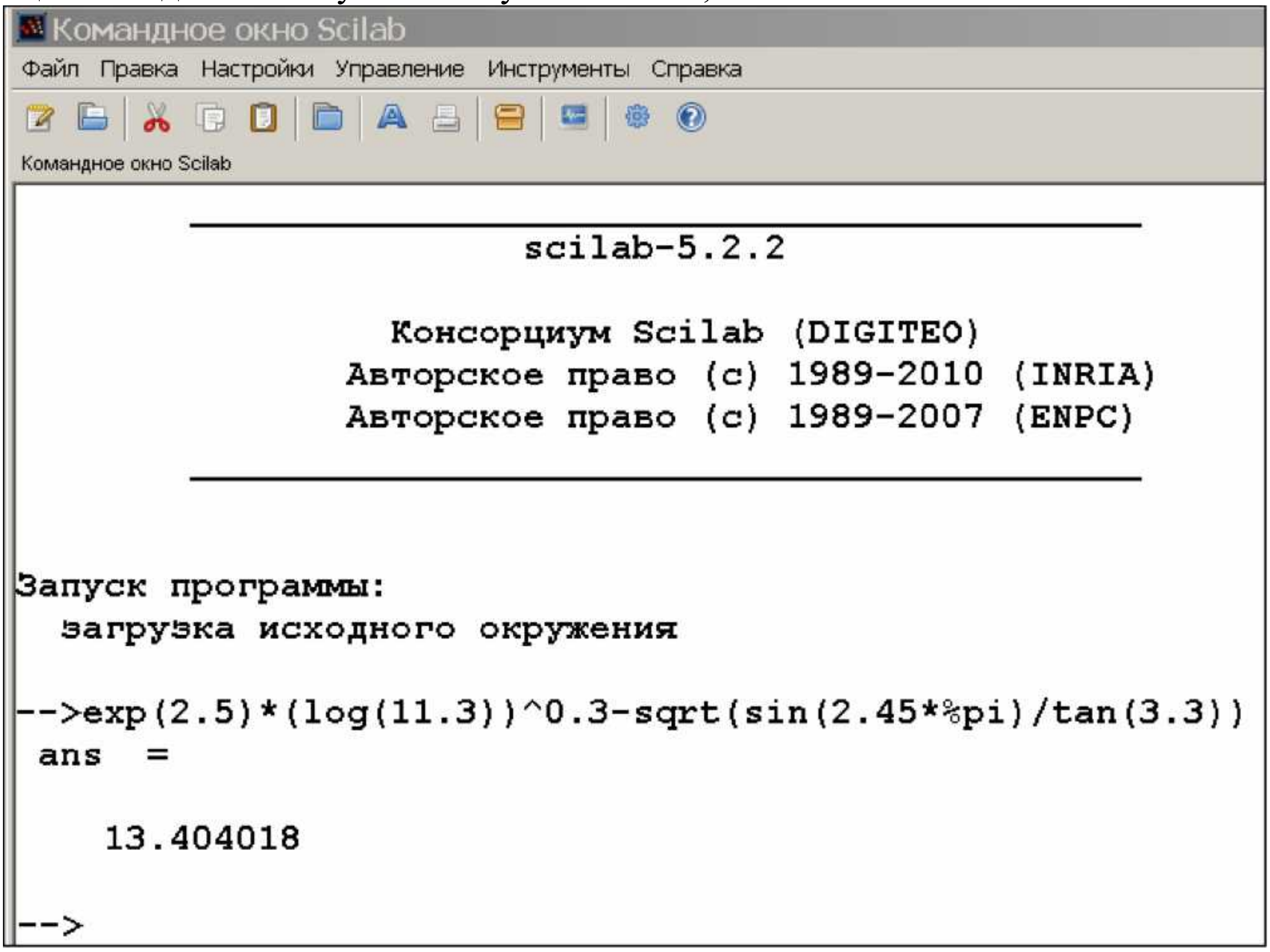



Рис. 1. Командная строка SciLab

Арифметические операции выполняются в обычном порядке: свойственном языкам программирования:

– действия в скобках;
– вычисление функции;
– возведение в степень ^;
– умножение * и деление слева направо / (5/2=2.5) и справа налево \ (5\2=0.4);
– сложение и вычитание +, -.

Для изменения порядка вычислений используйте скобки.

Переменные в Scilab. Любая переменная до использования в формулах и выражениях должна быть определена. Для этого используется оператор присваивания «=», который в общем виде записывается

Имя переменной = Значение выражения

Действие оператора: в переменную, имя которой указано слева, будет записано значение выражения, указанного справа.

Примечание 1. Имя переменной не должно совпадать с именами встроенных процедур, функций и встроенных переменных системы и может содержать до 24 символов.

Примечание 2. Система различает большие и малые буквы в именах переменных, т.е. ABC, abc, Abc, aBc - это имена разных переменных.

Примечание 3. Выражение в правой части оператора присваивания может быть числом, арифметическим выражением, строкой символов или символьным выражением. Если переменная является символьной, то выражение в правой части оператора присваивания следует брать в одинарные кавычки.

Примечание 4. Если для сохранения результата операции переменная пользователем не назначена, то SciLab определяет временную переменную *ans*, которую можно использовать в дальнейших вычислениях.

Системные переменные в SciLab начинаются с символа %:

%i - мнимая единица;

%pi - число пи (3.1415926);

%e - экспонента 1 (2.7182818);

%inf - машинный символ бесконечности;

%NaN - неопределенный результат (0/0 и т.п.);

%eps - условный ноль (2.220E-16)

Вывод в Scilab. По умолчанию результат выводится с восемью значащими цифрами после запятой. Для контроля вывода применяют команду printf с заданным форматом, который соответствует правилам, принятым для этой команды в языке C (рис. 2).

Текущий документ, отражающий работу пользователя с системой Scilab, содержащий строки ввода, вывода и сообщения об ошибках, принято называть сессией. Значения всех переменных, вычисленные в течение текущей сессии, сохраняются в специально зарезервированной области памяти, называемой рабочим пространством системы. Определения всех переменных и функций, входящих в текущую сессию можно сохранить в виде файла, саму сессию сохранить нельзя.

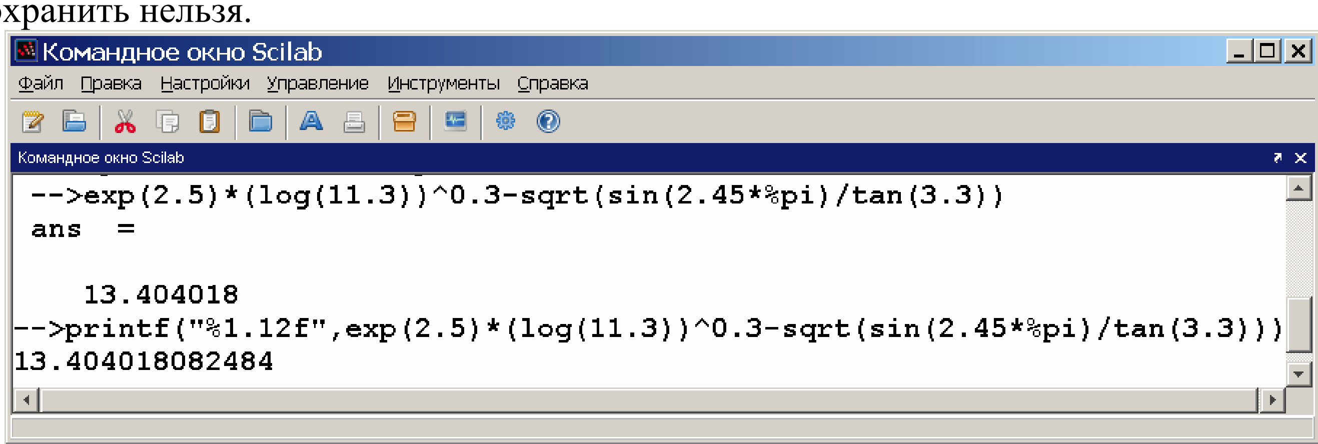


Рис. 2. Форматированный вывод (12 знаков после запятой)

**Редактирование и отладка файлов-сценариев**

Файл-сценарий - это список команд Scilab, сохраненный на диске. Для подготовки, редактирования и отладки файлов-сценариев служит специальный редактор SciPad, который можно вызвать, нажав кнопку на панели инструментов (рис. 3). В результате работы этой команды будет создан новый файл-сценарий. По умолчанию он имеет имя Untitledl. sce.

Окно редактора файлов-сценариев выглядит стандартно, т.е. имеет заголовок, меню, панели инструментов, строку состояния. Ввод текста в окно редактора файла-сценария осуществляется по правилам, принятым для команд Scilab.

Для сохранения введенной информации необходимо выполнить команду File →Save As... (Файл →Схранить как...) из меню редактора. Файлы-сценарии сохраняют с расширением . sce.

Открыть ранее созданный файл можно с помощью команды главного меню File → Open (Файл→ Открыть).

Выполнить операторы файла-сценария можно из меню редактора SciPad с помощью команды Execute → Load into Scilab (Выполнение →Загрузить в Scilab).

’V Console

1Ш

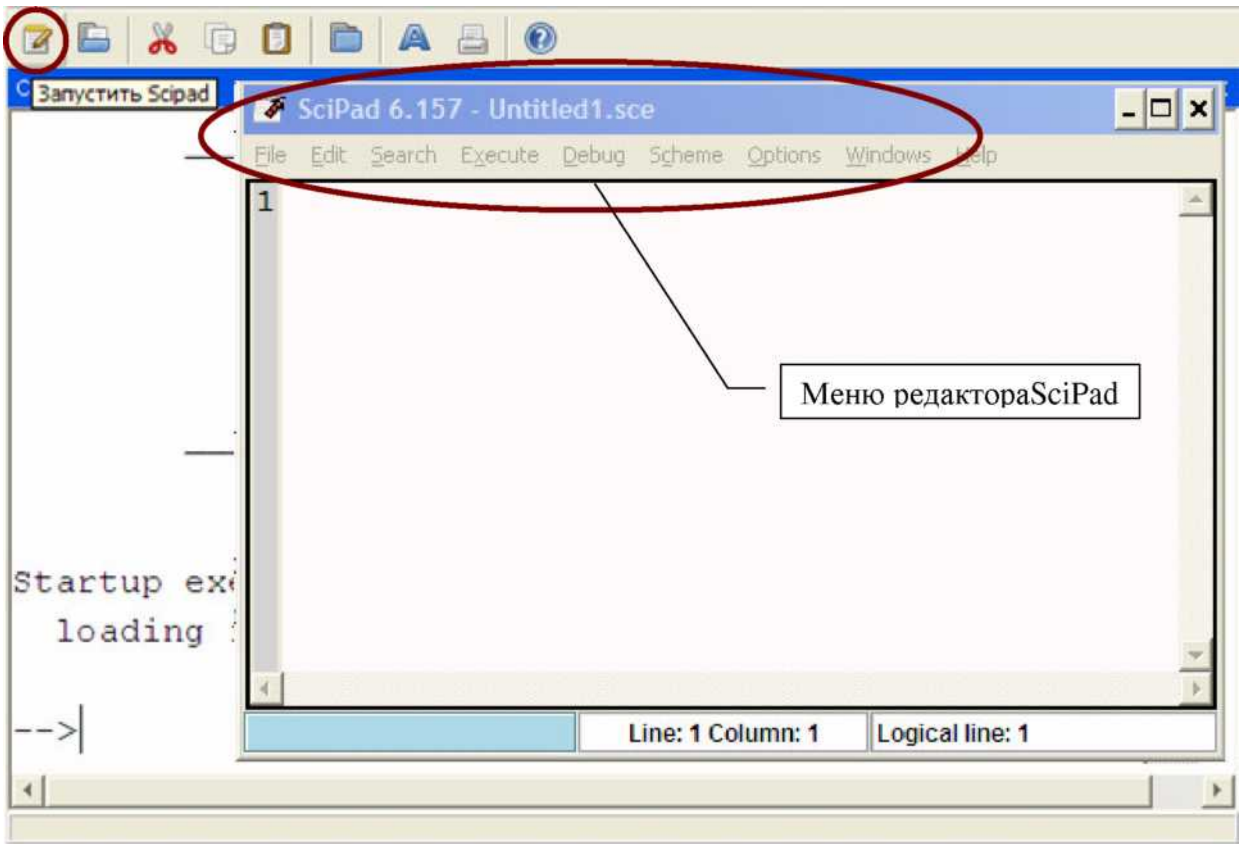



Рис. 3. Редактор SciPad

Примечание 1. Точка с запятой «;» ставится после тех команд, которые не требуют вывода значений.

Примечание 2. Строка после символов // не воспринимается как команда - это текстовый комментарий.

**Встроенные функции**

В общем виде обращение к функции в Scilab имеет вид:

имя_переменной = имя_функции(пер_1 [, пер_2, ...])

- имя_переменной - переменная, в которую будут записаны результаты работы функции: этот параметр может отсутствовать, тогда значение, вычисленное функцией, будет присвоено системной переменной *ans;*
- имя_функции - имя встроенной функции или ранее созданной пользователем;
- пер_1, пер_2,... - список аргументов функции.

В табл. 1 приведены наиболее часто используемые элементарные математические функции.

Табл. 1. Элементарные математические функции

| Функция | Описание функции | Функция | Описание функции |
|---|---|---|---|
| Тригонометрические | | | |
| sin(x) | синус числа x | asin(x) | арксинус числа x |
| cos(x) | косинус числа x | acos(x) | арккосинус числа x |
| tan(x) | тангенс числа x | atan(x) | арктангенс числа x |
| cotg(x) | котангенс числа x | | |
| Экспоненциальные | | | |
| exp(x) | экспонента числа x | log(x) | натуральный логарифм числа x |
| Другие | | | |
| sqrt(x) | корень квадратный из числа x | log10(x) | десятичный логарифм от числа x |
| abs(x) | модуль числа x | log2(x) | логарифм по основанию два от числа x |

**Функции, определенные пользователем**

Функция - это именованная логически законченная группа команд, которую можно вызывать для выполнения по имени, и предназначена для неоднократного использования. Функция имеет входные параметры и не выполняется без их предварительного задания. Задать вид функции можно с помощью конструкции function ... endfunction:

function[имя1,...,имяN] = имя_функции
(переменная_1,...,переменная_M)
тело функции
endfunction

- имя1, имяN- список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений;
- имя_функции - имя, с которым эта функция будет вызываться;
- переменная_1,…, переменная_M - входные параметры.

Все имена переменных внутри функции, а также имена из списка входных и выходных параметров воспринимаются системой как локальные, т.е. считаются определенными только внутри функции.

Пример. Вычислить площадь четырехгранника, если даны длины его ребер. Решение приведено на рис. 4.

```
-->function [r]=area(a,b,c)
-->  p=(a+b+c)/2;
-->  //Формула Герона
-->  r=sqrt(p*(p-a)*(p-b)*(p-c));
-->endfunction

-->S=area(2,3,3)+area(3,2,3)+area(3,3,2)+area(2,2,2)
 S  =

    10.217332
```

Рис.4

**Массивы в Scilab**

Массив - пронумерованная совокупность однородных данных, состоящая из фиксированного числа элементов, обозначенная одним именем. Доступ к отдельным элементам массива осуществляется по целочисленному индексу, то есть по номеру элемента в массиве. В зависимости от количества индексов, определяющих положение элемента в массиве, массивы разделяют на одномерные (вектора- строки, вектора-столбцы), двумерные (матрицы) и многомерные.

SciLab представляет все данные в виде массивов, даже переменная - это двумерный массив с размерностью один на один.

Работа с векторами. Вектора - это одномерные (линейные) массивы, в которых позиция каждого элемента задается единственным числом - его номером. При задании векторов элементы разделяются пробелами, запятой (,) или точкой с запятой (;):

a1=[3 4 9 2] - вектор-строка

a1=[3, 4, 9, 2]- вектор-строка

a1=[3; 4; 9; 2] - вектор-столбец.

Доступ к элементам вектора осуществляется заданием его индекса в круглых скобках после имени.

Если значения элементов вектора являются арифметической прогрессией, то их можно задать с помощью операции «:». С помощью функции length можно определить, сколько элементов попало в вектор.

Пример. Сформировать одномерный массив чисел в диапазоне от 3.7 до 8.947 с приращением 0.3. Решение приведено на рис. 5.

```
-->Mas=3.7:0.3:8.947
 Mas  =

         column 1 to 8

   3.7    4.    4.3    4.6    4.9    5.2    5.5    5.8

         column  9 to 16

   6.1    6.4    6.7    7.    7.3    7.6    7.9    8.2

         column 17 to 18

   8.5    8.8

-->length(Mas)
 ans  =

   18.
```

Рис. 5. Формирование одномерного массива

Поэлементные операции с векторами

Чтобы выполнить поэлементное умножение, деление, возведение в степень векторов, используют следующие знаки «.*», «./», «.^» (без пробелов!). В результате выполнения этих операций получается тоже вектор.

Примечание. Умножение и деление вектора на число выполняется с помощью обычных знаков «*» «/» без точки.

Пример. Протабулировать функцию $y(x) = e^{-x} \sin 10x$ на отрезке [0,1] с шагом 0.05.

Для решения задачи необходимо вначале задать вектор, содержащий значения аргумента X, а затем вычислить элементы вектора функции Y, используя поэлементное умножение!!! Решение представлено на рис. 6.

```
-->x=[0:0.2:1]
 x  =
   0.    0.2    0.4    0.6    0.8    1.
-->y=exp(-x).*sin(10*x)
 y  =
         column 1 to 4

   0.    0.7444698  - 0.5072999  - 0.1533465

         column 5 to 6

   0.4445473  - 0.2001342
```

Рис. 6. Табулирование функции

**Работа с матрицами**

Для хранения матриц в системе SciLab используются двумерные массивы, имеющие уникальное имя. Доступ к элементам массива осуществляется при помощи двух индексов: номера строки и номера столбца, указанных в круглых скобках, например C( 2,3).

Матрицы можно ввести либо по строкам

```
a = [3 1 -1; 2 4 3]
```

либо по столбцам

```
a = [[3;2] [1;4] [-1;3]]
```

Для того чтобы узнать размеры двумерного массива и «геометрию» векторов (вектор-столбцы или вектор-строки), нужно использовать функцию size:

size(a)
 ans =
2 3

Заданная матрица a содержит две строки и три столбца.

Примечание. Над массивами одинаковых размеров допускаются операции сложения (+) и вычитания (-). Для поэлементного перемножения используется знак «.*». Для поэлементного деления массивов - знаки «./» и «\.». Для поэлементного возведения в степень - знак «.^».

Привычные знаки «*», «/» предназначены в системе SciLab для матричных операций.

Транспонирование матрицы, так же как и векторов производится с помощью символов «.’».

**Построение двумерных графиков**

В двумерных графиках положение каждой точки задается двумя величинами (координатами по оси X и по оси Y), поэтому прежде, чем строить график необходимо:

1) сформировать массив x;

2) создать массив y, вычислив значение функции для каждого значения массива x.

Использование функции plot. Обращение к функции имеет вид:

*plot(x,y,[xcap, ycap, caption]),*

где

*x* - массив абсцисс;

*y* - массив ординат;

*xcap*, *ycap*, *caption* - подписи осей *X*, *У* и графика соответственно.

Пример. Построить график функции *y = sin(cos(x))* на интервале *[-2π; 2 π]* с шагом 0,1.

Необходимо:

1) сформировать массив *x*;

2) создать массив *y*, вычислив значение функции для каждого значения массива *x*;

3) построить график функции с помощью функции plot (рис. 7).

| | |
|---|---|
| `-->x=[-2*%pi:0.1:2*` | `-->x=[-2*%pi` |
| `-->y=sin(cos(x));` | `-->y=sin(cos` |
| `-->plot(x,y)` | `-->plot(x,y)` |

Рис. 7

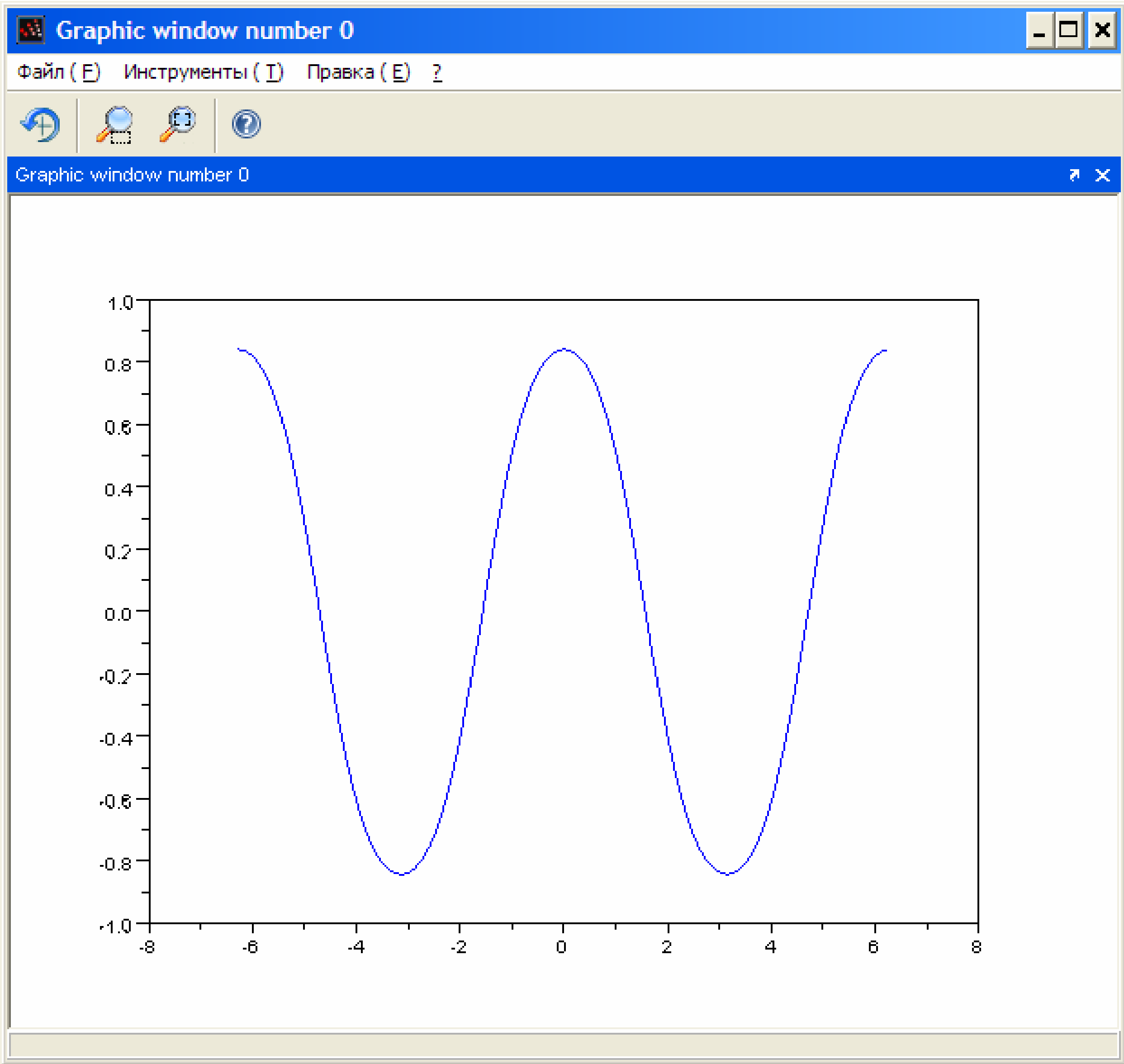



Рис. 8. Построенный график функции

**Форматирования графика.** В Scilab внешний вид графика можно изменять, используя дополнительный параметр функции plot: строку из трех символов, заключенных в апострофы. Эти символы задают цвет линии, тип маркера и тип линии соответственно (табл. 2).

Табл. 2. Некоторые символы для задания внешнего вида функции

| Цвет | | Маркер | | Линия | |
|---|---|---|---|---|---|
| Символ | Цвет | Символ | Тип маркера | Символ | Тип линии |
| r | красный | * | звездочка | - | сплошная |
| g | зеленый | s | квадрат | -.. | пунктир две точки |
| b | синий | d | ромб | -. | пунктир точка |
| k | черный | o | кружок | - - | пунктирная |

Построение пунктирного графика красного цвета с маркерами в виде *:

```
-->plot(x,y,'r*--')
```

Внешний вид графика можно также изменять, используя функцию *xgrid()*, чтобы задать линии сетки и функцию *xtitle()*, чтобы задать название графика и его осей:

```
-->plot(x,y,'r')
-->xgrid();
-->xtitle('sin(cos(x))','X','Y')
```

Расположение осей графика можно изменить, получив доступ к параметрам осей с помощью команды *gca()* и используя соответствующие команды. Например, для перемещения оси X в начало координат следует использовать:

```
-->a=gca();
-->a.x_location="origin";
```

На рис. 9 приведен отформатированный график.

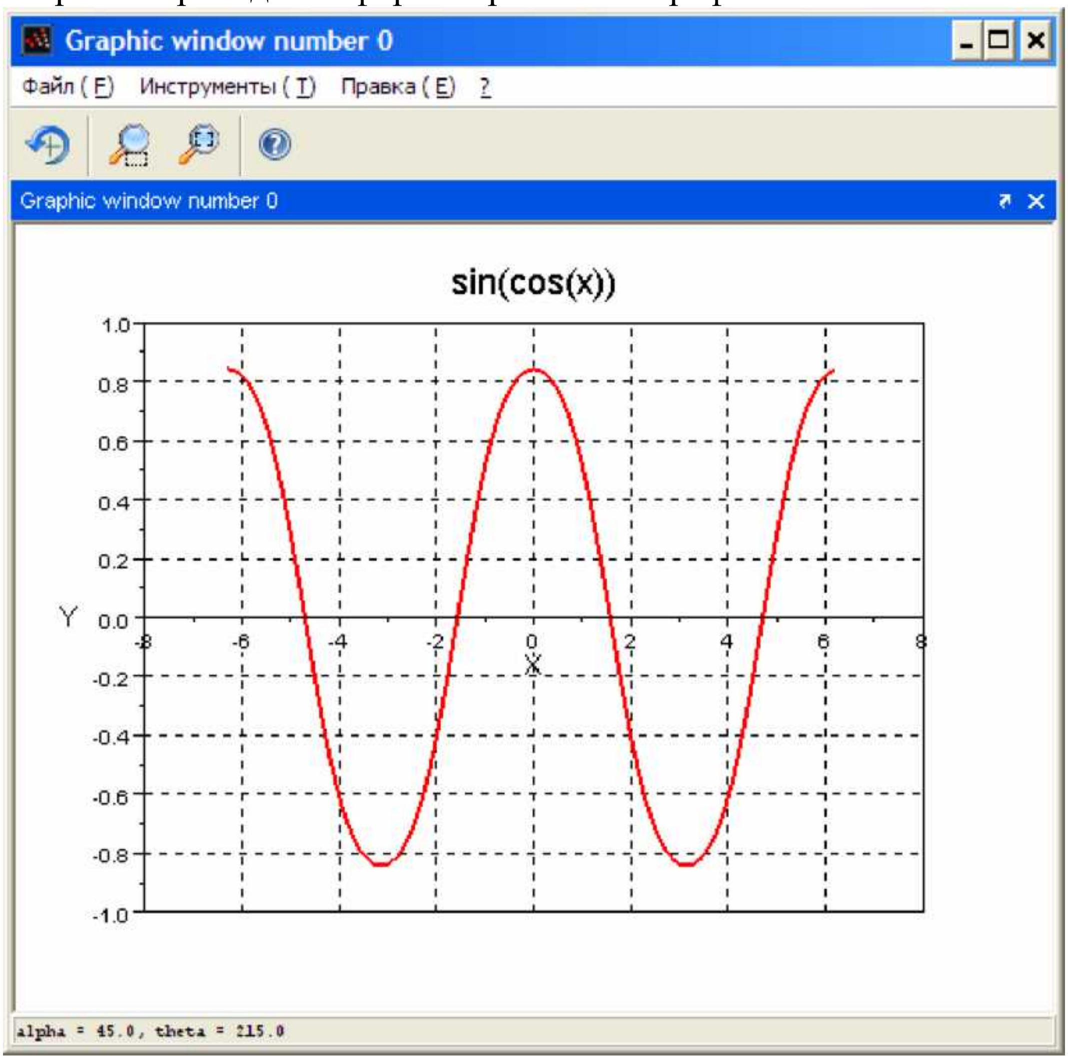



Рис. 9. Отформатированный график

На рис. 10 приведен файл-сценарий построения графика рассматриваемой функции.

```
//Задание значений аргумента
x=[-2*%pi:0.1:2*%pi];
//Вычисление значений функции
y=sin(cos(x));
//Построение графика (цвет красный)
plot(x,y,'r')
//
//Нанесение на график сетки
xgrid();
//Задание заголовков графика и осей
xtitle('sin(cos(x))','X','Y')
//доступ к параметрам осей графика
a=gca();
//Расположение оси X в начале координат
a.x_location="origin";
```

Рис. 10 Файл-сценарий построения графика рассматриваемой функции.

## Лабораторная работа №2

Тема: **Решение нелинейных уравнений и систем.**

### 1. Цель работы

Использование методов решения нелинейных уравнений и систем для решения конкретных производственных задач.

### 2. Учебные вопросы, подлежащие рассмотрению:

– Отделение корней уравнения.

– Решение нелинейных уравнений:

✓ метод половинного деления,

✓ метод простых итераций,

✓ метод Ньютона.

### 3. Порядок выполнения работы

**Задание 1.**

Вычислить наименьший положительный корень заданного уравнения с точностью $\varepsilon=10^{-3}$. Работу провести в три этапа:

1) Провести графическое отделение корней уравнения.

2) Сузить отрезок, полученный графическим способом до отрезка длиной 0.1.

3) Вычислить приближенное решение методом половинного деления.

По итогам выполнения заданий представить корень уравнения, вычисленный с указанной точностью.

Номер варианта соответствует порядковому номеру в списке.

**Задание 2.**

Найти решение уравнения с точностью $\varepsilon=10^{-4}$, используя метод простой итерации и один из методов Ньютона.

Номер варианта соответствует порядковому номеру в списке.

*Исполнение*: Освоить реализацию итерационных процессов с использованием логических функций в MS Excel.

*Оценка*: Использование инструментальных пакетов для решения трансцендентных уравнений.

Методические указания

**Краткая теория метода половинного деления**

Многие проблемы физики, строительной и технической механики, техники и других областей приводят к задаче нахождения корней нелинейного уравнения с одной переменной. В общем случае нелинейное уравнение можно записать в виде

$$F(x) = 0, \qquad (1)$$

где функция $F(x)$ определена и непрерывна на конечном или бесконечном интервале $[a, b]$. Примерами трансцендентных уравнений являются уравнения $x^2 - \sin x = 0$; $\operatorname{tg}(x - 6) = 3^x$.

Определение 1. Число $x^*$, такое, что $F(x^*)=0$ называется корнем уравнения (1).

Для подавляющего числа нелинейных уравнений вида (1) невозможно (или очень сложно) решить задачу нахождения корней уравнения аналитическими методами. Поэтому на практике такие уравнения решаются численными методами, вместо точного решения $x^*$ вычисляется приближенное решение.

Определение 2. Число $\tilde{x}$, такое, что $|x^* - \tilde{x}| \le \varepsilon$ называется приближенным решением уравнения (1), найденным с точностью $\varepsilon > 0$.

Задача численного нахождения приближенных корней состоит из двух этапов: отделение корней, то есть нахождение достаточно малого интервала $(a, b)$, в котором содержится один корень уравнения (1), и уточнение корня, т.е. вычисление приближенного решения с необходимой точностью.

*Графическое отделение корней.*

Для того чтобы провести графическое отделение корней, надо построить график функции $y = F(x)$ и визуально определить интервал, на котором находится ровно один корень уравнения.

*Уточнение корня методом половинного деления.*

Основная идея нахождения приближенного решения заключается в сокращении первоначального интервала, определенного при графическом отделении, до интервала длиной $2\varepsilon$. После того, как удалось сократить интервал до заданной величины, можно определить $\tilde{x} = (a + b) / 2$. В этом случае условие $|x^* - \tilde{x}| \le \varepsilon$ выполнено.

В методе половинного деления сокращение интервала происходит делением отрезка $[a, b]$ пополам и выбора той из половин, которой принадлежит искомый корень уравнения.

Итак, алгоритм численного решения уравнения (1) методом половинного деления заключается в выполнении следующих шагов:

1. определить начальный отрезок $[a, b]$;
2. найти точку с – середину отрезка $[a, b]$, $c = (a + b) / 2$;
3. проверить, какому из отрезков $[a, c]$ или $[c, b]$ принадлежит корень. Легко видеть, что проверка выполняется так:

если $f(a)*f(c)<,0$, то корень принадлежит отрезка $[a, c]$ и в дальнейшем надо положить $b = c$, в противном случае корень принадлежит отрезку $[c, b]$ и следует положить $a = c$;

4. если длина отрезка [$a$, $b$] больше 2 $\varepsilon$, то перейти к пункту 2;

5. закончить вычисления, положив $\tilde{x}$ =($a$ +$b$) / 2.

Теория электронных таблиц

Для реализации данного алгоритма необходимо воспользоваться логическими функциями. Логические функции предназначены для проверки выполнения условия или для проверки нескольких условий. В отличие от математических функций, при проведении вычислений с логическими функциями мы оперируем понятиями ИСТИНА и ЛОЖЬ.

В общем виде функция, позволяющая учесть при вычислениях условия выглядит так:

ЕСЛИ(*логическое выражение; значение1; значение2*)

*логическое выражение* – это выражение, принимающее значения ИСТИНА или ЛОЖЬ. Например, С15=1 – это логическое выражение. Если значение в ячейке С15 равно 1, то выражение принимает значение ИСТИНА, иначе – ЛОЖЬ.

*Значение1* – это значение, которое заносится в ячейку, если *логическое выражение* равно ИСТИНА; *значение2* – это значение, которое заносится в ячейку, если *логическое выражение* равно ЛОЖЬ.

До семи функций ЕСЛИ могут быть вложены друг в друга в качестве значений аргументов *значение1* и *значение2* для конструирования более сложных проверок. Если любой из аргументов функции ЕСЛИ является массивом, все элементы массива вычисляются при выполнении функции ЕСЛИ.

Microsoft Excel предлагает дополнительные функции, которые можно применять для анализа данных с использованием условий. Например, для вычисления числа появлений текстовой строки или числа в диапазоне ячеек используется функция СЧЁТЕСЛИ. Для вычисления суммы значений, попадающих в интервал, заданный текстовой строкой или числами, используется функция СУММАЕСЛИ.

Для записи логических выражений (условий) служат стандартные операции сравнения: = (равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), <> (не равно). Если требуется записать более сложные условия, включающие в себя несколько простых условий, то приходится применять такие логические функции, как И, ИЛИ, НЕ.

И(*логическое выражение1; логическое выражение2*; ...)

Функция И будет иметь значение ИСТИНА, если все логические выражения, перечисленные в скобках, истинны. В противном случае результатом функции И будет значение ЛОЖЬ. Всего можно указать до 30 различных условий.

ИЛИ(*логическое выражение1; логическое выражение2*; ...)

Функция ИЛИ возвращает значение ИСТИНА, если хотя бы один из аргументов имеет значение ИСТИНА и возвращает ЛОЖЬ, если все аргументы имеют значение ЛОЖЬ.

НЕ(логическое выражение)

Если логическое выражение имеет значение ЛОЖЬ, то функция НЕ возвращает значение ИСТИНА; если *логическое выражение* имеет значение ИСТИНА, то функция НЕ возвращает значение ЛОЖЬ.

*Методика выполнения*

1) Запустить программу Excel. Создать документ Книга1. Сохранить в отведенное место на жестком диске под оригинальным именем, например, Книга Иванова. Первая лабораторная работа выполняется на листе Лист1 вашей первой электронной книги, переименовать его в Лаб1.

2) Создать заголовок сверху по центру будущей таблицы, используя шрифты помельче, 12 пунктов, Arial курсив и покрупнее, 14 пунктов, Times Roman полужирный. Выделить название темы лабораторной цветом.

3) Сделать подпись Выполнил, Дата, Вариант. Дата должна показывать текущую дату в заданном полном формате. Вставить свою функцию по варианту в математической нотации как объект MS Equation. Выделить формулу цветной рамкой.

4) Завести отдельные ячейки для начала интервала $a_x$ и шага табулирования $h_x$, ячейки D7 и D8 на рисунке 1.1. Начало интервала определить вручную. Для заполнения диапазона ячеек со значениями аргумента функции использовать абсолютные адреса этих ячеек. В ячейки C7 и C8 вставьте пояснения Начало интервала $a_x$ =.. и Шаг $h_x$ =….

5) В строке 10 оформить шапку таблицы: Номер точки, Значение аргумента, Значение функции, и ниже, в строке 11 сокращенно: №, x, y.

6) Заполнение таблицы.

Создание столбца номеров точек. Набрать 1 в B12. Протащить маркер заполнения формул с нажатым Ctrl до 79 в всплывающем окне для получения арифметической прогрессии с приращением 1 в диапазоне ячеек B13:B90.

Создание столбца *x*. В первую сверху ячейку столбца, C12 набираем ссылку на начало интервала =$D$7 (абсолютный адрес). Под ней в ячейку C13 формулу для расчета следующего значения аргумента через предыдущее =C12+$D$8 (комбинация абсолютных и относительных адресов). Протаскиваем маркер заполнения через диапазон ячеек C14:C90 для получения в нем формул расчета значений аргумента.

Создание столбца *y*. В ячейку D12 набираем формулу расчета функции по варианту для первой строки таблицы. Для 0-го варианта =sin(3*C12)^2-log10(C12-2) (относительный адрес x). Далее используем технику протаскивания формулы для заполнения диапазона ячеек D13:D90 со значениями функции.

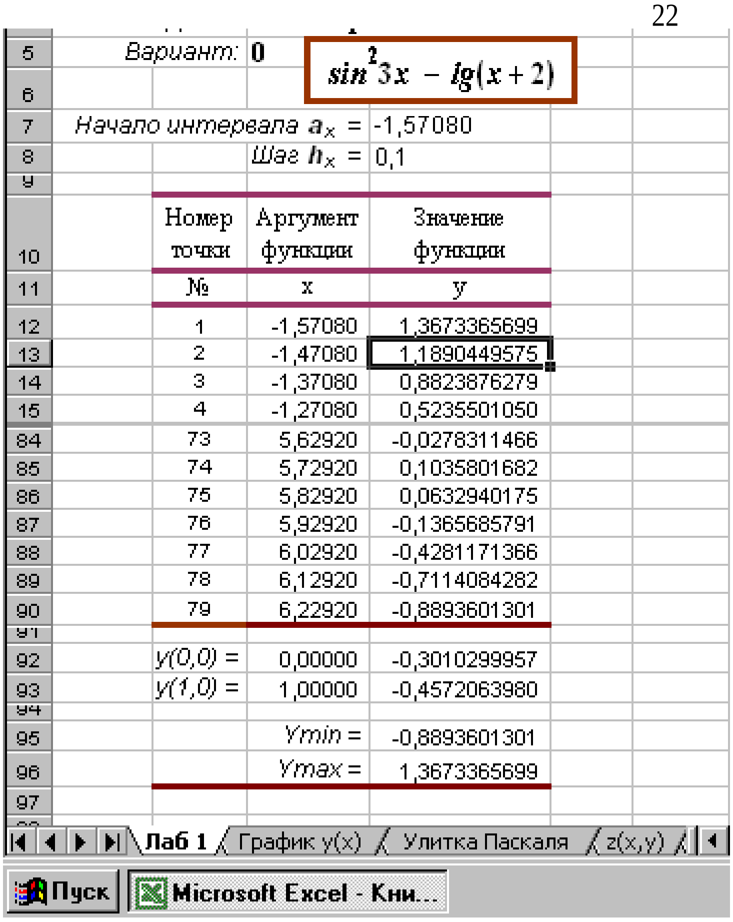
7) Форматирование таблицы. Вывести 5 значащих цифр для значений *x* и 10 для *y*.

Выровнять по десятичной точке. Назначить холодный цвет, например, синий, отрицательным значениям и красный положительным. Навести рамки в таблице по образцу на рисунке 1.1.

8) Посчитать значения функции для $x$=0.0 и $x$=1.0.

9) Посчитать максимальное $Y_{max}$ и минимальное $Y_{min}$ значения функции в полученном диапазоне ячеек.

1) График функции в Excel, как и все остальные диаграммы, строится по дискретным значениям. Исходные данные для графика получены в предыдущей лабораторной работе.



| | | | |
|---|---|---|---|
| 5 | Вариант: | 0 | $sin^2 3x - lg(x+2)$ |
| 7 | Начало интервала $a_x$ = | -1,57080 | |
| 8 | Шаг $h_x$ = | 0,1 | |
| 10 | Номер точки | Аргумент функции | Значение функции |
| 11 | № | x | y |
| 12 | 1 | -1,57080 | 1,3673365699 |
| 13 | 2 | -1,47080 | 1,1890449575 |
| 14 | 3 | -1,37080 | 0,8823876279 |
| 15 | 4 | -1,27080 | 0,5235501050 |
| 84 | 73 | 5,62920 | -0,0278311466 |
| 85 | 74 | 5,72920 | 0,1035801682 |
| 86 | 75 | 5,82920 | 0,0632940175 |
| 87 | 76 | 5,92920 | -0,1365685791 |
| 88 | 77 | 6,02920 | -0,4281171366 |
| 89 | 78 | 6,12920 | -0,7114084282 |
| 90 | 79 | 6,22920 | -0,8893601301 |
| 92 | y(0,0) = | 0,00000 | -0,3010299957 |
| 93 | y(1,0) = | 1,00000 | -0,4572063980 |
| 95 | | Ymin = | -0,8893601301 |
| 96 | | Ymax = | 1,3673365699 |



Рисунок 1.1. Образец оформления лабораторной работы

2) Запускаем Мастера диаграмм. Выбираем тип диаграммы График. Во втором окне мастера указываем исходные данные для первого ряда Значения: столбец *y*, Подписи оси X:

столбец *x*, Имя: набираем с клавиатуры Грубый график. Для второго ряда Значения и Подписи по оси *X* те же, Имя: Сглаженный график. Лишние ряды удалить.

2) Отформатировать график по образцу рисунка 2.1.

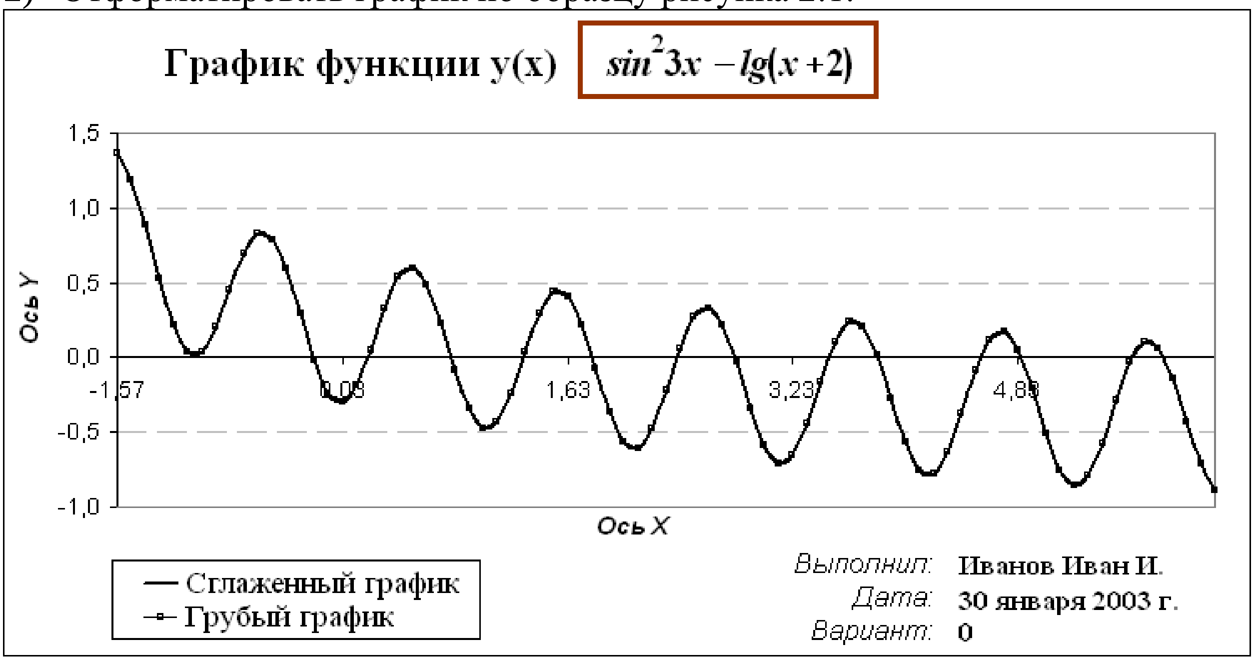




Рисунок 2.1. Образец оформления графика

На грубом графике соединить точки отрезками прямых сплошными, тонкими, синего цвета, в точках синие прозрачные квадратные маркеры 5 пунктов. На глаженном графике -соединяющие точки линии сплошные, толстые, красные, сглаженные, без маркеров.

Настроить порядок рядов так, чтобы тонкие синие линии были видны на фоне красных.

Оформить график заголовком, формулу скопировать на график. Подписать оси как Ось *X* и Ось *Y*. Ось *X* провести через *y*=0. Проредить деления шкал, в подписях делений отобразить 2-3 цифры, чтобы не загромождали график. Настроить диапазон оси *Y* от $Y_{min}$ до $Y_{max}$ своей функции. Подпись Выполнил Дата Вариант подставить в шесть надписей из соответствующих ячеек листа. Выровнять и распределить надписи инструментами панели Рисование – Действия – Выровнять, Распределить, Группировка. Убрать серую заливку фона области построения. Линии сетки оставить только горизонтальные по основным делениям оси *Y*, уменьшить их яркость до светло-серого цвета.

1) Работа с книгой. Перенести диаграмму на отдельный лист под именем График $y(x)$. Для обзора точности построения ломаной и сглаженной линий отобразить график в максимальном масштабе.

2) Создать новый лист, назвать его «Корень».

3) Оформить рабочий лист, написав заголовок «Решение уравнения методом половинного деления», ниже заголовка добавить надпись «Решить уравнение <Ваше уравнение>».

4) Подготовьте лист для проведения графического отделения корней. Сделайте надпись «1. Графическое отделение корней», в ячейку A6 сделайте поясняющую надпись «a=», в ячейку A7 – надпись «b=», в ячейку A8 – «Шаг=».

5) По графику, построенному в листе Графика, определите отрезок, на котором находится наименьший положительный корень. Левую границу отрезка укажите в ячейке B6, а его правую границу – в ячейке B7. В ячейку B8 занесите формулу вычисления шага $(b\text{-}a)/10$.

6) Протабулируйте функцию $y = f(x)$ на интервале $[a, b]$ с вычисленным шагом $h$.

Постройте график данной функции.

7) Сделайте надпись «2. Уточнение отрезка». Подготовьте ячейки для занесения величин $a, b, h$ (см. п.3). Запишите значения $a$ и $b$, определенные с помощью графика, величину шага укажите равной 0,1.

8) Протабулируйте функцию $y = f(x)$ на новом интервале $[a, b]$ с заданным шагом $h=0{,}1$.

9) Сделайте надпись «3. Метод половинного деления». Аналогично тому, как это было сделано в пп. 3 и 6, подготовьте ячейки для занесения величин $a, b$. По таблице значений функции $y = f(x)$ определите отрезок, на котором функция меняет знак. Это и будет первоначальный отрезок $[a, b]$ для метода половинного деления. Скопируйте значения границ этого отрезка в соответствующие ячейки.

10) Ниже границ отрезка укажите заданное значение точности ε, внесите поясняющую надпись «Epsilon=».

11) Оформите вычисления по методу половинного деления в виде таблицы:

| № шага | a | b | c | f(a) | f(b) | f(c) | b-a | решение |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | | | | | | | | |
| 2 | | | | | | | | |

Установите вывод 7 значащих цифр после десятичной запятой для всех значений в столбцах 2-9.

На первом шаге в ячейках столбцов 2 (значение *a*) и 3 (значение *b*) поставьте ссылки на ячейки, содержащие границы отрезка после его уточнения (п.8). В столбец 3 внесите формулу, соответствующую вычислению *c=(a+b)/2*, в столбцы 5, 6 и 7 запишите формулы для вычисления *f(a), f(b)* и *f(c)*. В столбце 8 вычислите длину отрезка [*a, b*], а в столбец 9 внесите формулу, реализующую условие ЕСЛИ $b\text{-}a \le 2\varepsilon$, ТО решение=с, ИНАЧЕ решение=” ”.

На втором шаге вычислений формулы в столбцах 4-9 переносятся (“растягиваются”) из первого шага. В столбец 2 вносится формула вида

ЕСЛИ(*f(a)*f(c)<=0;a;c*), а в столбец 3 вносится аналогичная формула

ЕСЛИ(*f(a)*f(c)>=0;b;c*).

Здесь вместо *f(a), f(c), a, b, c* указываются ссылки на соответствующие ячейки. Дальнейшие шаги выполняются так же, как второй шаг.

Копируйте строки таблицы до тех пор, пока в столбце 9 не появится значение искомого корня.

12) Скопируйте полученное значение приближенного решения в отдельную ячейку, слева от нее введите надпись «Ответ:», установите вывод 3 значащих цифр после запятой.

Контрольные вопросы и задания

**Варианты функции *y*(*x*) к Заданию 1**

Таблица 1

| № варианта | Функция $y(x)$ |
|---|---|
| 0 | $\sin^2 3x - \lg(x+2)$ |
| 1 | $\sin^2\left(\frac{x}{2}+2\right)^2 - \frac{x}{5}$ |
| 2 | $0.2\,x - \cos^2 3x$ |
| 3 | $\log_2(x+4) - \frac{1}{2}\left(\sin\left(\frac{x+6}{3}\right)^3 + 5\right)$ |
| 4 | $\frac{\sin(x-8)^2}{3} - \frac{x-2}{5}$ |
| 5 | $\frac{\cos(x-8)^2}{3} + \ln(x+3) - 1.5$ |
| 6 | $\frac{\cos(x-8)^2 - \sin(x+1)}{2} - 0.2$ |
| 7 | $\frac{1}{2}\sin\frac{(\boldsymbol{x}+5)^2}{2} + \left(\frac{\boldsymbol{x}-2}{4}\right)^2$ |
| 8 | $\sin\,(x^2+3) + \frac{\sin^2 x}{2}$ |
| 9 | $\frac{2\sin 8x}{3} + \frac{3\cos x}{4}$ |
| 10 | $\cos 5x\ \ln\frac{x+3}{2}$ |
| 11 | $\frac{3}{\boldsymbol{x}+3}\sin\left(\frac{2\boldsymbol{x}+6}{3}\right)^2$ |
| 12 | $\cos\frac{(x+2)^2}{2}\ \ln\left(\frac{x+3}{2}\right)$ |
| 13 | $\frac{\cos\left((x-5)^2+1\right)}{2}\ln(x+2.5)$ |
| 14 | $\frac{\sin(x-1)^2}{e^{x/4}} - 0.1$ |
| 15 | $\frac{1}{2}\sin\frac{(x+3)^2}{2}\ln(x+2)$ |

| № варианта | Функция $y(x)$ |
|---|---|
| 16 | $\frac{1}{2}\sin\frac{(x+3)^2}{2}+\frac{\ln(x+2)}{2}-1$ |
| 17 | $\frac{\sin\left(2(x-2)^2\right)}{e^{x/4}}-0.1x$ |
| 18 | $\frac{\sin 10x}{e^{\frac{(x-2)^2}{5}}}+0.5$ |
| 19 | $\frac{\sin(10x-20)}{e^{\frac{(x-2)^2}{5}}}+0.25x$ |
| 20 | $\frac{\cos(5x+10)}{\ln(4x+8)}$ |
| 21 | $\frac{2\cos 5x}{3}-\frac{x}{6}+\frac{1}{3}$ |
| 22 | $2\sqrt{2\cos^2(\frac{x}{2}+2)^2+2}-\frac{x}{5}-3$ |
| 23 | $\frac{\sin(10x-20)}{e^{\frac{(x-2)^2}{5}}}-0.25x$ |
| 24 | $\frac{1}{2}\cos\left(\frac{x}{3}\right)-\sin^2(3x)$ |
| 25 | $\sin\left(\frac{x}{3}\right)-\cos^2(3x)$ |

**Варианты функции *y*(*x*) к Заданию 2**

| N | *f(x)* | [*a*, *b*] |
|---|---|---|
| 1 | $(\sin x)^2 - \frac{5}{6}\sin x + \frac{1}{6}$ | $[0,1]$ |
| 2 | $(\sin x)^2 + \frac{7}{12}\sin x + \frac{1}{12}$ | $[-1,0]$ |
| 3 | $(\sin x)^2 - \frac{1}{30}\sin x - \frac{1}{30}$ | $[-0.5,0.5]$ |
| 4 | $(\cos x)^2 + \frac{2}{35}\cos x - \frac{1}{35}$ | $[0,2]$ |
| 5 | $(\cos x)^2 - \left(\frac{1}{\sqrt{2}} + \frac{1}{4}\right)\cos x + \frac{1}{4\sqrt{2}}$ | $[0,1.5]$ |
| 6 | $(\cos x)^2 + \frac{1}{2}\cos x + \frac{1}{18}$ | $[0,2]$ |
| 7 | $(\ln x)^2 - 5\ln x + 6$ | *[5,25]* |
| 8 | $(\ln x)^2 - \ln x - 2$ | *[0.1,10]* |
| .9 | $(\ln x)^2 - \frac{3}{4}\ln x + \frac{1}{8}$ | *[0.1,2]* |
| 10 | $(tgx)^2 + (\sqrt{3} - 1)tgx - \sqrt{3}$ | $[-1.2,1]$ |
| 11 | $(tgx)^2 - \frac{28}{9}tgx + \frac{1}{3}$ | $[0,1.5]$ |

| | | |
|---|---|---|
| 12 | $(tgx)^2 - \frac{53}{6} tgx - \frac{3}{2}$ | $[-0.5, 1.5]$ |
| 13 | $x^4 - 7x^2 + 10$ | *[0,3]* |
| 14 | $x^4 - \frac{10}{3} x^2 + 1$ | *[0,2]* |
| 15 | $x^4 - \frac{13}{2} x^2 + 3$ | *[0,3]* |
| 16 | $(\sin x)^2 + \frac{5}{6}\sin x + \frac{1}{6}$ | $[-1, 0]$ |
| 17 | $(\sin x)^2 - \frac{7}{12}\sin x + \frac{1}{12}$ | $[0, 1]$ |
| 18 | $(\sin x)^2 + \frac{1}{30}\sin x - \frac{1}{30}$ | $[-0.5, 0.5]$ |
| 19 | $(\cos x)^2 - \frac{2}{35}\cos x - \frac{1}{35}$ | $[0, 3]$ |
| 20 | $(\cos x)^2 + \left(\frac{1}{\sqrt{2}} - \frac{1}{4}\right)\cos x - \frac{1}{4\sqrt{2}}$ | $[0, 2]$ |
| 21 | $(\cos x)^2 - \frac{1}{2}\cos x + \frac{1}{18}$ | $[0, 2]$ |
| 22 | $(\lg x)^2 + \frac{5}{3}\lg x - \frac{2}{3}$ | *[0.001,3]* |
| 23 | $(\lg x)^2 - \lg x - \frac{3}{4}$ | *[0.1,35]* |
| 24 | $(\lg x)^2 + \frac{3}{4}\lg x - \frac{1}{4}$ | *[0.01,3]* |
| 25 | $(tgx)^2 - (1 + \frac{1}{\sqrt{3}}) tgx + \frac{1}{\sqrt{3}}$ | $[0, 1]$ |

**Задание 3**

Найти решение уравнения $x^3+x-1=0$ с точностью $\varepsilon=0.001$ методом деления отрезка пополам. Для решения данной задачи написать программу на языке программирования VBA.

Решение: Работа выполняется в следующей последовательности:

Запустить VISUAL BASIC. В окне New Project выбрать Стандартный.EXE вкладки New (новый) и кликнуть по кнопке «Открыть».

На экране появится новый проект и форма Project1 - Form1 (Form), содержание которой нужно заполнить необходимыми компонентами, с помощью которых можно будет выполнить необходимые вычисления.

Во-первых, необходимы элементы, которые будут использоваться для ввода данных *A*, *B* и *E*. Используем для этой цели элементы Text1, Text2 и Text3. Во-вторых, необходимы элементы, в которых будут отражены результаты вычисления $x_n$ и $y_n$. Используем для этой цели элементы Text4 и Text5.

Для оформления надписей для обозначения текстовых окон Text1, Text2, Text3, Text4 и Text5 можно использовать метки Label1, Label2, Label3, Label4 и Label5 соответственно.

Чтобы процедура пользователя была доступна на всех формах приложения, ее текст можно записать в специальном окне Module - модуль приложения. Программный код такой программы будет храниться в файлах приложения с расширением *.bas.

```
Function fnf(x As Single) As Single = x ^ 3 + x -1
End Function
```

Для выполнения функций вычисления необходима одна кнопка Command1. Это будет «Старт». Вторая кнопка Command2 выполняет функцию завершения работы приложения - кнопка «Финиш».

Теперь следует все указанные выше компоненты вынести на форму. Для этого на панели стандартных элементов находим значок одного из компонент, активизируем его и «переносим» на форму: рисуем условный прямоугольник в том месте формы, где будет предположительно находиться этот элемент, учитывая при этом его размеры.

Аналогичные действия выполняем поочередно для каждого элемента, пока на форме не появятся все 12 элементов.

Далее изменяем свойства объектов, используя страницу свойств Properties. Для изменения свойств элементов следует активизировать элемент - выделить его на форме. Далее для выделенного элемента в окне свойств можно вносить изменения: для Command1 изменить свойство Caption = ‘Старт‘, для Command2 - свойство Caption = ‘Финиш‘. Для остальных элементов задать значения свойств:

Label1.Caption = ‘Введите значение A‘

Label2.Caption = ‘Введите значение B‘

Label3.Caption = ‘Введите значение E‘

Label4.Caption = ‘Вывод значения xn‘

Label5.Caption = ‘Вывод значения yn‘

Заготовку формы для примера с измененными свойствами смотрите на рис. 1.

Для создания программного кода в проводнике проекта или в меню View выбрать команду Code.

В окне Project1 - Form1 (Code) выбрать в списке General название кнопки 1 и кликнуть по ней. В результате появится заготовка записи программного кода для Command1 с инициированием для нее события Click. Значения A, B, E пользователь должен ввести в окна Text1, Text2, Text3. Но в текстовом окне данные получаются также текстовые, типа String. Поэтому используем функцию Val( ) для перевода в числовой тип Single, который объявлен для переменных A, B, E в начале программы в операторе Dim. Для вывода значений xn, yn используется функция Round( ), которая округляет количество знаков после запятой до четырёх.

Аналогичные действия выполняем в отношении кнопки 2. Для Command2 будет выполняться одно действие - закрыть приложение. Это выполняет команда End.

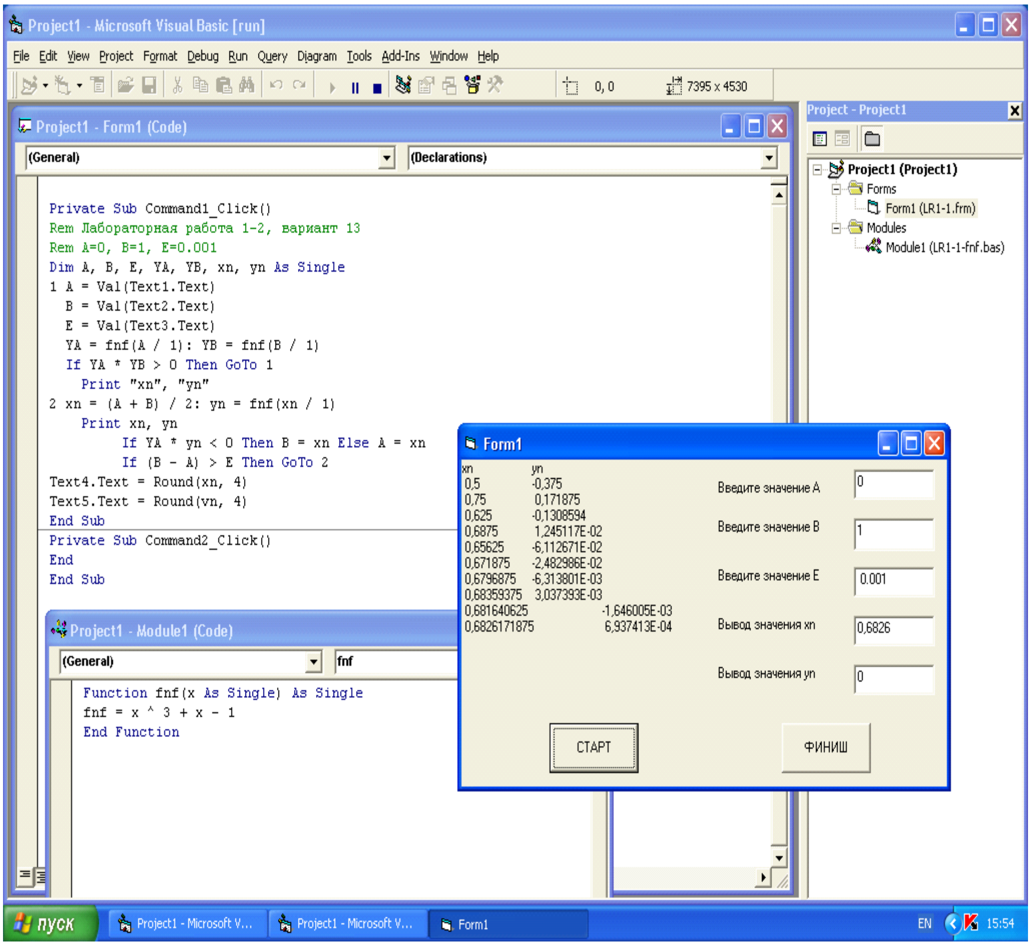



Рис. 1. Заготовка формы для примера с измененными свойствами.

Программный код для примера нахождения корней уравнения методом деления отрезка пополам будет иметь вид:

```
Private Sub Command1_Click()
Rem Лабораторная работа 1, вариант ? A=0, B=1, E=0.001
Dim A, B, E, YA, YB, xn, yn As Single
A= Val(Text1.Text)
B= Val(Text2.Text)
E= Val(Text3.Text)
```

YA= fnf(A / 1): YB = fnf(B / 1)

If YA * YB > 0 Then GoTo 1"xn", "yn"

xn = (A + B) / 2: yn = fnf(xn / 1)xn, ynYA * yn < 0 Then B = xn Else A = xn(B - A) > E Then GoTo 2.Text = Round(xn, 4).Text = Round(vn, 4)SubSub Command2_Click()

End Sub

. Для сохранения проекта надо выполнить команду меню File (файл) → Save Project As (Сохранить проект как). В диалоговом окне Save File As (Сохранить файл как) выбрать название диска, на котором должна быть создана папка для хранения файлов нового проекта. Далее для сохранения формы, следует набрать имя файла в поле ввода File Name (Имя файла), например Primer1-1, и нажать клавишу «Enter». Форма Form1 будет сохранена в файле с расширением Primer1-1.frm.

После сохранения формы следует сохранить проект. В VISUAL BASIC это происходит автоматически, после чего должно появиться диалоговое окно Save Project As -Сохранить проект. Следует набрать то же имя Primer1-2 и нажать клавишу «Enter». Файл проекта будет сохранен с расширением Primer1-1.vbp.

Теперь запустить проект на выполнение. Выбираем команду Run → Start или Shift+F5. В поле Text1 нужно ввести значение А, в поле Text2 - значение В, в поле Text3 - значение Е - после чего нажать кнопку «Старт». В окнах Text4, Text5 появились значения xn, yn. Для завершения работы приложения следует нажать кнопку «Финиш».

Для решения трансцендентных уравнений в Scilab применяют функцию

*fsolve(x0,f),*

где *x0* - начальное приближение;

*f* - функция, описывающая левую часть уравнения *f(x) = 0*. Последовательно вызывая функцию *fsolve* с различными начальными приближениями, получают все решения заданного уравнения на заданном диапазоне.

Пример. Найти корни уравнения $\frac{e^x}{5} - 2 \cdot (x-1)^2 = 0$

Этап 1. Отделение корней. С помощью конструкции function ... endfunction определяется вид функции, задается вектор диапазона значений аргумента x, вычисляется вектор значений функции y и строится график функции (рис. 11).

```
-->function [y]=F(x)
-->y=exp(x)/5-2*(x-1).^2
-->endfunction

-->x=[-1:0.01:6];

-->y=F(x);

-->plot(x,y)
```

Рис. 11. Команды для задания вида функции и построения графика функции

Этап 2. По графику функции (рис. 12) определяются начальные приближения корней (0, 2, 5). Уточнить корни можно: либо последовательно вызывая функцию fsolve с различными начальными приближениями (рис. 13); либо задав вектор начальных приближений, тогда fsolve вызывается один раз (рис. 14).

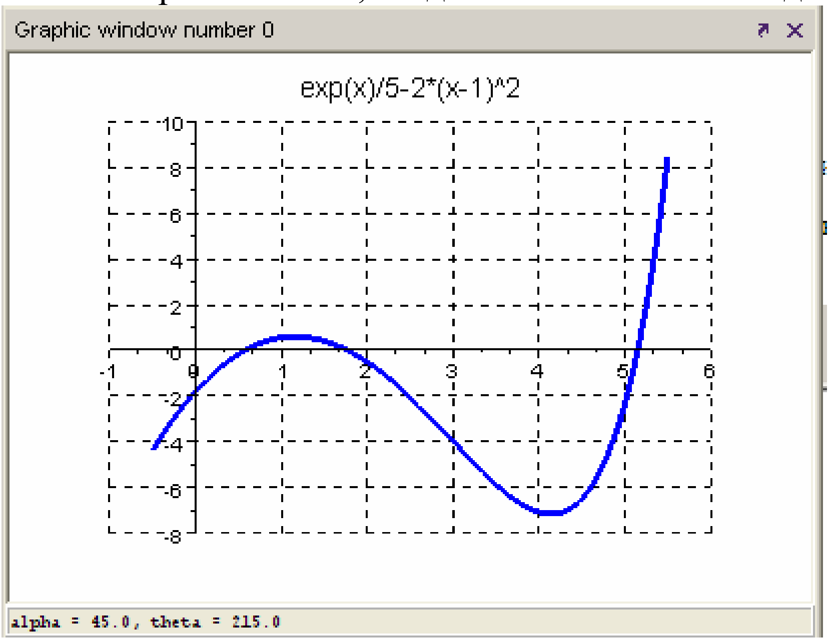



Рис. 12. Графическое решение трансцендентного уравнения

```
-->X(1)=fsolve(0,F); X(2)=fsolve(2,F); X(3)=fsolve(5,F);

-->X
 X  =

    0.5778406
    1.7638701
    5.1476865
```

Рис. 13. Последовательный вызов fsolve с различным начальным приближением

```
-->X=fsolve([0;2;5],F)
 X  =

    0.5778406
    1.7638701
    5.1476865
```

Рис. 14. Вызов fsolve при задании начальных приближений в виде вектора

На рис. 15 приведен файл-сценарий нахождения корней трансцендентного уравнения.

```
//Задание вида функции
function [y]=F(x)
y=exp(x)/5-2*(x-1).^2
endfunction
//
//Задание значений аргумента
x=[-0.5:0.01:5.5];
//
//Вычисление значений функции
y=F(x);
//
//Изменение параметров графика
a=gca();
a.x_location="origin";
a.y_location="origin";
xgrid();
xtitle('exp(x)/5-2*(x-1)^2')
//
//Построение графика функции
plot(x,y)
//
//Нахождение трех корней транцендентного
//уравнения при их заданных начальных
//приближениях 0 2 5
fsolve([0;2;5],F)
```

Рис. 15. Файл-сценарий построения графика трансцендентного уравнения и нахождения его корней

### Контрольные вопросы.

1) Какие точные методы решения нелинейных уравнений вы знаете?
2) Для чего нужен первый этап - отделение корней?
3) Сформулируйте условия существования решения уравнения. Являются ли эти требования необходимыми и достаточными?
4) Что можно сказать о точности методов половинного деления, хорд, касательных и комбинированного? По каким параметрам их еще можно сравнить?

## Лабораторная работа №2

Тема: **Численные методы решения систем линейных уравнений.**

1. **Цель работы**

Использование методов решения систем линейных уравнений для решения конкретных научных задач.

2. **Учебные вопросы, подлежащие рассмотрению:**

– Метод простой итерации
– Оценка погрешности решения системы линейных алгебраических уравнений.
– Понятие об обусловленности. Метод прогонки, трехдиагональная матрица.

– Релаксация.

3. **Порядок выполнения работы**

*Задание*

Дана система трех линейных уравнений с тремя неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

1. Решите систему методом Гаусса:

а) используя «ручную» схему единственного деления, двумя способами: без перестановки строк; с перестановкой строк; расчеты выполняйте с тремя знаками после запятой (с применением калькулятора); подставьте найденные решения в исходную систему, вычислите невязки и сравните полученные решения; выбрав ведущие элементы схемы единственного деления, найдите значение определителя системы;

б) с помощью программы для ЭВМ с пооперационным учетом ошибок.

2. Решите систему методом простой итерации с точностью $\varepsilon = 10^{-4}$ с помощью программы для ЭВМ.

*Исполнение*: применить а) метод Гаусса; б) метод простой итерации используя любой инструментальный пакет.

*Оценка*: Сопоставление полученных результатов, решаемых различными методами.

Методические указания

В табличном процессоре Excel для решения систем уравнений есть два варианта: реализация алгоритмов в электронной таблице с помощью основных средств табличного процессора и использование специальных средств.

Первый вариант проиллюстрирован на примере системы уравнений

$$\begin{cases} x_1 + 2x_2 + 2x_4 = 5 \\ 2x_1 + 4x_2 - x_3 + 5x_4 = -1 \\ x_1 + 3x_2 + 5x_4 = -3 \\ 3x_1 + 7x_2 - 3x_3 + 9x_4 = -13 \end{cases}$$

На рис. приведены идентичные тексты в Excel, но один — в режиме отображения формул, а другой — значений. Из них прекрасно видно устройство алгоритма.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Решение системы линейных алгебраических уравнений Ax=b | | | | |
| 2 | методом Гаусса | | | | |
| 3 | A | | | | b |
| 4 | | | | | |
| 5 | 1 | 1 | 0 | 2 | 5 |
| 6 | 2 | 4 | -1 | 5 | -1 |
| 7 | 1 | 3 | 0 | 5 | -3 |
| 8 | 3 | 7 | -3 | 9 | -13 |
| 9 | Первое преобразование | | | | |
| 10 | 1 | =B5/$A$5 | =C5/$A$5 | =D5/$A$5 | =E5/$A$5 |
| 11 | 0 | =B6-B5*$A6 | =C6-C5*$A6 | =D6-D5*$A6 | =E6-E5*$A6 |
| 12 | 0 | =B7-B5*$A7 | =C7-C5*$A7 | =D7-D5*$A7 | =E7-E5*$A7 |
| 13 | 0 | =B8-B5*$A8 | =C8-C5*$A8 | =D8-D5*$A8 | =E8-E5*$A8 |
| 14 | Второе преобразование | | | | |
| 15 | 1 | 1 | 0 | 2 | 5 |
| 16 | 0 | 1 | =C11/$B$11 | =D11/$B$11 | =E11/$B$11 |
| 17 | 0 | 0 | =C12-C11*$B$12/$B$11 | =D12-D11*$B$12/$B$11 | =E12-E11*$B$12/$B$11 |
| 18 | 0 | 0 | =C13-C11*$B$13/$B$11 | =D13-D11*$B$13/$B$11 | =E13-E11*$B$13/$B$11 |
| 19 | Третье преобразование | | | | |
| 20 | 1 | 1 | 0 | 2 | 5 |
| 21 | 0 | 1 | -0,5 | 0,5 | -5,5 |
| 22 | 0 | 0 | 1 | =D17/$C$17 | =E17/$C$17 |
| 23 | 0 | 0 | 0 | =D18-D17*$C$18/$C$17 | =E18-E17*$C$18/$C$17 |
| 24 | Четвертое преобразование | | | | |
| 25 | 1 | 1 | 0 | 2 | 5 |
| 26 | 0 | 1 | -0,5 | 0,5 | -5,5 |
| 27 | 0 | 0 | 1 | 2 | 3 |
| 28 | 0 | 0 | 0 | 1 | =E23/D23 |
| 29 | Обратный ход | | | | |
| 30 | x4= | =E28 | | | |
| 31 | x3= | =E27-D27*B30 | | | |
| 32 | x2= | =E26-D26*B30-C26*B31 | | | |
| 33 | x1= | =E25-D25*B30-C25*B31-B25*B32 | | | |

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Решение системы линейных алгебраических уравнений Ax=b | | | | |
| 2 | методом Гаусса | | | | |
| 3 | A | | | | b |
| 4 | | | | | |
| 5 | 1 | 1 | 0 | 2 | 5 |
| 6 | 2 | 4 | -1 | 5 | -1 |
| 7 | 1 | 3 | 0 | 5 | -3 |
| 8 | 3 | 7 | -3 | 9 | -13 |
| 9 | Первое преобразование | | | | |
| 10 | 1 | 1 | 0 | 2 | 5 |
| 11 | 0 | 2 | -1 | 1 | -11 |
| 12 | 0 | 2 | 0 | 3 | -8 |
| 13 | 0 | 4 | -3 | 3 | -28 |
| 14 | Второе преобразование | | | | |
| 15 | 1 | 1 | 0 | 2 | 5 |
| 16 | 0 | 1 | -0,5 | 0,5 | -5,5 |
| 17 | 0 | 0 | 1 | 2 | 3 |
| 18 | 0 | 0 | -1 | 1 | -6 |
| 19 | Третье преобразование | | | | |
| 20 | 1 | 1 | 0 | 2 | 5 |
| 21 | 0 | 1 | -0,5 | 0,5 | -5,5 |
| 22 | 0 | 0 | 1 | 2 | 3 |
| 23 | 0 | 0 | 0 | 3 | -3 |
| 24 | Четвертое преобразование | | | | |
| 25 | 1 | 1 | 0 | 2 | 5 |
| 26 | 0 | 1 | -0,5 | 0,5 | -5,5 |
| 27 | 0 | 0 | 1 | 2 | 3 |
| 28 | 0 | 0 | 0 | 1 | -1 |
| 29 | Обратный ход | | | | |
| 30 | x4= | -1 | | | |
| 31 | x3= | 5 | | | |
| 32 | x2= | -2,5 | | | |
| 33 | x1= | 9,5 | | | |

Рис.1

Второй вариант не столь очевиден. Среди встроенных в Excel математических программ программы решения систем уравнений, строго говоря, нет.

Неточно так же, как для решения уравнений было использовано средство Подбор параметра, для решения систем может быть использовано средство, предназначенное совсем для другой цели (для решения задач оптимизации).

Это средство - Поиск решения. Поясним его использование и приведем примеры применения.

Средство Поиск решения активируется в меню Сервис (если это средство не установлено, то это необходимо сделать).

Предварительно проводят следующую подготовительную работу.

Отводят для каждой переменной по ячейке.

Вводят формулы для вычисления правых частей уравнений системы (по одной формуле в ячейку).

После этого запускают Поиск решения. Возможный вид экрана для подготовки к решению той же системы, что и выше, приведен на рис.

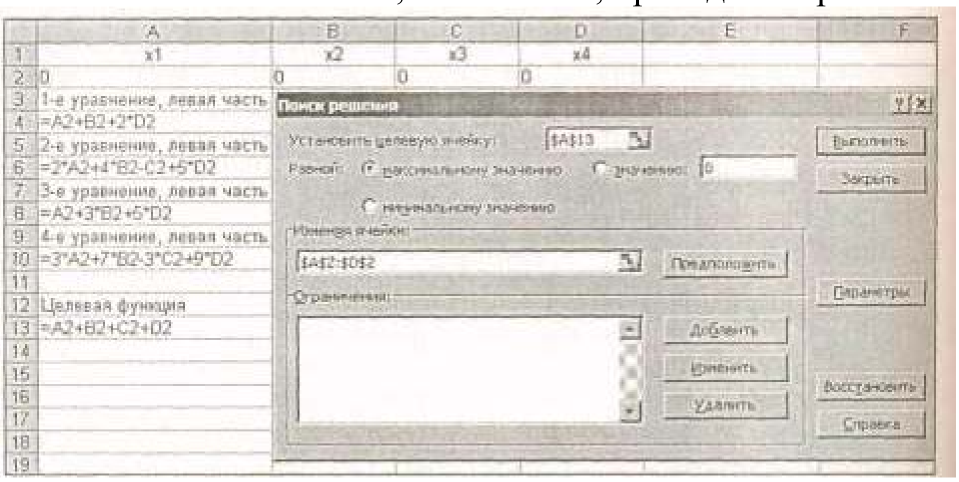


Рис.2

На этом рисунке под переменные отведены ячейки A2:D2, под формулы — ячейки A4, A6, A8 и A10. Какие именно числовые значения переменных будут введены в ячейки A2:D2, при решении системы линейных уравнений значения не имеет (итерационная процедура, заложенная в Поиск решения, стартует с этих значений).

Еще один элемент в таблице — целевая функция. Она в данном случае особой роли не играет, но какую-нибудь формулу ввести необходимо, иначе Поиск решения работать не может (напомним, что эта программа нацелена на другой класс задач). Точно так же неважно, как установлен флажок: «максимальному значению» или «минимальному значению».

Теперь необходимо ввести то, что в форме на рис. 2 именуется ограничениями.

Щелкнув по кнопке Добавить, получают другую форму (рис.3 ).


Рис3.

В форму, изображенную на рис. 3. надо ввести четыре условия (по числу уравнений системы). На рисунке отражено последнее условие. Ссылка на ячейку A10 обусловлена тем, что в ней -формула для левой части 4-го уравнения, знак '=' выбран из меню, число (-13) введено с клавиатуры (правая часть 4-го уравнения). После ввода последнего ограничения нажимают кнопку ОК и возвращаются в основную форму Поиск решения. Щелкнув по кнопке Выполнить, получают результат (рис. 4).


Рис4.

Каким методом это решение получено, можно узнать, щелкнув по кнопке Параметры (см. рис. 2). Там, в частности, есть знакомый нам метод Ньютона (наверняка сильно видоизмененный, поскольку применяется на самом деле к решению гораздо более сложной задачи).

Поиск решения можно попытаться применить и к решению систем нелинейных уравнений. В этом случае выбор начального приближения очень важен, в зависимости от него решение может быть получено или не получено и могут быть получены разные решения.

Контрольные вопросы и задания

1) Запустить программу Excel, открыть рабочую книгу лабораторных работ. Создать в ней новый лист, дать ему имя «Мет.итераций».

2) Оформить рабочий лист.

Значения коэффициентов матрицы $P$ и вектора $b$ взять по своему варианту, таблица 4 Приложения.

3) Привести систему к виду (4).

4) Умножить матрицу $P$ и вектор $b$ на транспонированную матрицу $P^T$

Преобразовать коэффициенты по формулам (4'). В формулах вычисления коэффициентов $a_{ij}$ и $b_i$ $(i \neq j)$ использовать абсолютную адресацию. Коэффициенты $a_{ii}$ задать равными нулю.

5) Для начала итерационного процесса задать значение вектора $x^0$ равным $b$.

Ввести формулы для вычисления $x^{k+1}$ и $d$. При ссылке на матрицу $A$ можно использовать абсолютные адреса, для вектора $b$ удобно ячейкам с элементами вектора присвоить имена и в формулах ссылаться на них:

$$x^{k+1} = \begin{array}{l} \text{=H24+v\_b1} \\ \text{=H25+v\_b2} \\ \text{=H26+v\_b3} \\ \text{=H27+v\_b4} \end{array}$$

Ячейкам со значениями величин $\varepsilon$ и $d$ также следует присвоить имена для использования в формулах.

Для переноса значений $x^{k+1}$ на следующий шаг в ячейки $x^k$ необходимо использовать функцию ЕСЛИ с условием $\varepsilon<d$.

6) Копировать ячейки, относящиеся к одному шагу итераций до тех пор, пока $\varepsilon<d$.

7) Выделить в отдельные ячейки окончательный ответ с 3 десятичными знаками. Сравнить полученные значения с точным решением системы уравнений и решением, полученным выше.

СЛАУ можно записать в матричном виде

$$Ax=b$$

где $A$ – матрица коэффициентов при неизвестных;

$x$ – вектор неизвестных;

$b$ – вектор свободных членов.

При решении СЛАУ можно использовать метод обратной матрицы. Тогда решение можно найти по формуле

$$x=A^{-1}\cdot b$$

Пример. Решить систему линейных алгебраических уравнений (СЛАУ)

$$\begin{cases} 6x_1 - 2x_2 + x_3 = 4 \\ 3x_1 + 2x_3 = 6 \\ 6x_1 - 2x_2 + 12x_3 = 1 \end{cases}$$

Для решения СЛАУ методом обратной матрицы необходимо:

1. Задать матрицу коэффициентов при неизвестных $A$ и вектор свободных членов $b$.
2. Проверить детерминант матрицы А.
3. Если детерминант не равен 0, найти обратную матрицу $A^{-1}$
4. Умножить обратную матрицу на вектор свободных членов.

Решение СЛАУ в системе Scilab.

Специальные матричные функции необходимые для решения СЛАУ методом обратной матрицы:

- Функция *det(A)* вычисляет определитель квадратной матрицы А.
- Функция *inv(A)* вычисляет обратную матрицу к матрице А.

Решение СЛАУ с помощью этого метода приведено на рис. 20.

```
-->A=[6 -2 1;3 0 2; 6 -2 12];

-->b=[4;6;1];

-->det(A)
 ans  =

    66.

-->A1=inv(A);

-->x=A1*b
 x  =

    2.1818182
    4.4090909
  - 0.2727273
```

Рис. 20. Решение СЛАУ с использованием обратной матрицы

Решить СЛАУ, заданную в виде уравнения $A\bar{x} - \vec{b} = 0$ используя функцию *linsolve(A, b)* можно, (рис. 21).

```
-->A=[6 -2 1;3 0 2; 6 -2 12];

-->b=[-4;-6;-1];

-->x=linsolve(A,b)
 x  =

    2.1818182
    4.4090909
  - 0.2727273
```

Рис. 21. Решение СЛАУ с использованием функции linsolve

Решите систему методом обратной матрицы.

Таблица 4 Варианты СЛАУ

| № варианта | Система линейных алгебраических уравнений | корни системы | № варианта | Система линейных алгебраических уравнений | корни системы |
|---|---|---|---|---|---|
| 1. | $\begin{cases} 3x-4y+4z-2k=4 \\ 6x+2y-3k=-5 \\ -9x+5y-2z+k=-2 \\ x-6y+z+3k=8 \end{cases}$ | 2<br>5<br>9<br>9 | 2. | $\begin{cases} 7x-2y+3z+5k=-15 \\ 3x+y-4z+3k=31 \\ x-4y+4z-k=-21 \\ x+y+4z-5k=-53 \end{cases}$ | -2<br>-4<br>-8<br>3 |
| 3. | $\begin{cases} x-6y+3z-k=-17 \\ 5y-3z+2k=15 \\ -4x+y+z-k=-6 \\ x+y-2z+3k=2 \end{cases}$ | 3<br>4<br>1<br>-1 | 4. | $\begin{cases} x-y-3z-4k=10 \\ -5x-3y-z-3k=-1 \\ 4x-3y+3z+9k=24 \\ 2x-y+5z+k=2 \end{cases}$ | 3<br>-5<br>-2<br>1 |
| 5. | $\begin{cases} 2x+3y-3z+5k=-2 \\ -2x+2y-z+4k=4 \\ 5x+y+4z+9k=-24 \\ -7x-9y-2z-7k=-8 \end{cases}$ | -2<br>4<br>0<br>-2 | 6. | $\begin{cases} -2x+4y-2z-4k=-4 \\ 2x+2y-z-k=9 \\ 4x-2y+3z+4k=0 \\ 5x+3y-z+3k=15 \end{cases}$ | 4<br>-2<br>-4<br>-1 |
| 7. | $\begin{cases} -x+4y+5z+3k=-2 \\ -2x-y-z-5k=-13 \\ 4x-y+z+k=-9 \\ 4x+3y-2z+k=7 \end{cases}$ | -2<br>1<br>-4<br>4 | 8. | $\begin{cases} 5x+y-2z-4k=2 \\ 3x-4y+3z-5k=-10 \\ -3x-4y+4z-5k=3 \\ -4x-y+5z-4k=-4 \end{cases}$ | -3<br>-1<br>-5<br>-2 |
| 9. | $\begin{cases} x+4y-z-5k=12 \\ -2x+5y-z+4k=-8 \\ -x+5y-4z-3k=0 \\ 2x+5y+2z+4k=11 \end{cases}$ | 4<br>1<br>1<br>-1 | 10. | $\begin{cases} 4x-3y-2z-3k=-12 \\ -5x+3y+3z+3k=8 \\ 2x-3y-5z-4k=0 \\ -2x+2y-4z-4k=4 \end{cases}$ | -1<br>1<br>-5<br>5 |
| 11. | $\begin{cases} 4x-4y-z+5k=-9 \\ -3x+4y-5z-4k=-12 \\ 3x-2y+z-k=6 \\ 2x-5y-5z-5k=3 \end{cases}$ | -1<br>-2<br>3<br>-2 | 12. | $\begin{cases} -3x-3y-5z+5k=7 \\ 2x+y+5z+k=-6 \\ -5x-2z-4z-5k=6 \\ -2x+y+2z-5k=3 \end{cases}$ | -3<br>4<br>-1<br>1 |
| 13. | $\begin{cases} x-2y+3z+2k=9 \\ x-2y-3z-k=9 \\ x-5y-2z-4k=1 \\ -4x+4y+2z+5k=-4 \end{cases}$ | 3<br>-2<br>-2<br>4 | 14. | $\begin{cases} -5x+2y+2z+4k=10 \\ 3x+3y-2z-2k=-1 \\ 2x+y-z+k=8 \\ 5x+3y+4z+k=21 \end{cases}$ | 2<br>1<br>1<br>4 |

| № варианта | Система линейных алгебраических уравнений | корни системы | № варианта | Система линейных алгебраических уравнений | корни системы |
|---|---|---|---|---|---|
| 15. | $\begin{cases} 3x+3y+4z-k=-33 \\ x-2y-z+5k=14 \\ 5x-2z+k=-10 \\ x-y+5z+5k=-3 \end{cases}$ | -3<br>-5<br>-2<br>1 | 16. | $\begin{cases} -4x+y-2z+2k=7 \\ -5x-y+3z-4k=20 \\ 2x-3y-2z+5k=4 \\ 5x-5y+3z-k=5 \end{cases}$ | -3<br>-3<br>2<br>1 |
| 17. | $\begin{cases} 4x+4y+5z+4k=5 \\ -5x+2y-z-2k=11 \\ -2x+3y-2z-7k=-9 \\ x-2y+z+5k=9 \end{cases}$ | -2<br>3<br>-3<br>4 | 18. | $\begin{cases} 4x-4y-2z+3k=13 \\ -5x-y+4z+3k=-8 \\ -4x-2y-5z+5k=-23 \\ 2x-2y-3z+4k=0 \end{cases}$ | 3<br>-2<br>2<br>-1 |
| 19. | $\begin{cases} 3x+3y-5z-k=11 \\ 4x-2y-z-5k=-5 \\ -3x-2y-5z+5k=7 \\ -4x-4y+2z+5k=-9 \end{cases}$ | -2<br>2<br>-2<br>-1 | 20. | $\begin{cases} -2x-2y-3z-2k=-9 \\ -x-3y+z-5k=7 \\ -x+y+5z-k=5 \\ 5x-5y-4z+5k=8 \end{cases}$ | 5<br>-2<br>3<br>-3 |
| 21. | $\begin{cases} -5x-2y+2z-k=-10 \\ -4x-4y+z+k=7 \\ 2x+y-4z-3k=-13 \\ -4x+3y-z+4k=0 \end{cases}$ | 2<br>-3<br>-1<br>4 | 22. | $\begin{cases} -5x-2y-2z+3k=6 \\ 3x-4y+3z+2k=-5 \\ -x+y-5z-k=-12 \\ -x+y-z-k=4 \end{cases}$ | -5<br>-2<br>4<br>-5 |
| 23. | $\begin{cases} 3x-y+2z-3k=13 \\ -2x+5y+3z+4k=-9 \\ -4x-5y-z+2k=-7 \\ 4x+3y+4z-5k=8 \end{cases}$ | 5<br>-2<br>1<br>2 | 24. | $\begin{cases} -2x+3y-2z+4k=2 \\ -4x-y+2z+4k=0 \\ -4x-5y-z+2k=-7 \\ 4x+3y+4z-5k=8 \end{cases}$ | -3<br>4<br>2<br>-3 |
| 25. | $\begin{cases} 3x-5y-z-4k=8 \\ -5x-4y+4z-2k=-24 \\ -2x-5y-z-4k=-2 \\ -4x+3y-2z-2k=-12 \end{cases}$ | 2<br>-2<br>-4<br>3 | 26. | $\begin{cases} 5x-y-3z+k=-7 \\ -4x+5y+4z-3k=0 \\ 3x-5y-4z+k=6 \\ x+y-z-3k=3 \end{cases}$ | -2<br>-2<br>-1<br>-1 |

## Контрольные вопросы

1. Основные понятия алгебры матриц. Действия с матрицами. Норма матрицы. Транспонированная и обратная матрицы.
2. Методы решения систем линейных алгебраических уравнений. Метод исключения Гаусса.
3. Вычисление обратной матрицы методом Гаусса.
4. Вычисление определителя методом Гаусса.
5. Итерационные методы решения систем линейных алгебраических уравнений. Метод Якоби. Условие сходимости итераций.
6. Итерационные методы решения систем линейных алгебраических уравнений. Метод Гаусса-Зейделя. Условие сходимости итераций.
7. Метод релаксации решения системы линейных алгебраических уравнений.

## Лабораторная работа №3

Тема: **Численное дифференцирование и интегрирование.**

1. **Цель работы**

Использование численных методов решения задач дифференцирования и интегрирования.

2. **Учебные вопросы, подлежащие рассмотрению:**

– Постановка задачи.
– Интерполяционный многочлен Лагранжа и интерполяционный многочлен Ньютона.
– Численное дифференцирование с помощью многочленов Лагранжа и Ньютона.
– Метод трапеции, метод Симпсона
– Численное интегрирование методами трапеции и Симпсона (парабол)

3. **Порядок выполнения работы**

*Задание 1*.

Вычислить значение производной функции, заданной таблично, используя интерполяционные формулы Лагранжа или Ньютона.

*Задание 2*.

Вычислить интеграл от заданной функции f(x) на отрезке [a,b] при делении отрезка на 10 равных частей следующими способами 1) по формуле трапеций; 2) по формуле Симпсона.

*Исполнение*: применить интерполяционные формулы Лагранжа или Ньютона используя любой инструментальный пакет для вычисления производной. Использовать формулы трапеции и формулы Симпсона для вычисления определенного интеграла.

*Оценка*: Сопоставление полученных результатов, решаемых различными методами.

Методические указания

Контрольные вопросы и задания

1. Вычислить значение производной функции, заданной таблично, используя интерполяционные формулы Лагранжа или Ньютона.

Вариант

| $X$ | $\sin x$ |
|---|---|
| 0,60 | 0,56464 |
| 0,65 | 0,60519 |
| 0,70 | 0,64422 |
| 0,75 | 0,68164 |
| 0,80 | 0,71736 |
| 0,85 | 0,75128 |
| 0,90 | 0,78333 |
| 0,95 | 0,81342 |
| 1,00 | 0,84147 |
| 1,05 | 0,86742 |
| 1,10 | 0,89121 |

Задание 2.

Вычислить интеграл от заданной функции f(x) на отрезке [a,b] при делении отрезка на 10 равных частей следующими способами 1) по формуле трапеций; 2) по формуле Симпсона.

Отрезок интегрирования разбивается на 10 равных частей. Для расчетов удобно составить единую таблицу значений по схеме:

| Xi | $y/2$ (1=0, 10) | $y_i$ (i=l, 2, 3, ..., 9) | $2y_i$ (i=1, 3, 5, 7, 9) |
|---|---|---|---|
| | | | |

По каждому из трех столбцов таблицы находятся суммы соответствующих значений подынтегральной функции (при этом по столбцу *у,- для* формулы трапеций находится сумма всех элементов столбца, а для формулы Симпсона — только с четными индексами).

Вариант

| Вариант | *m* | *a* | *b* |
|---|---|---|---|
| 1 | $0{,}37e^{\sin x}$ | 0 | 1 |
| 2 | $0,5+x\lg x$ | 1 | 2 |
| 3 | $(x+1{,}9)\sin(x/3)$ | 1 | 2 |
| 4 | $\frac{1}{x}\ln(x+2)$ | 2 | 3 |
| 5 | $\frac{3\cos x}{2x+1,7}$ | 0 | 1 |
| 6 | $(2x+0{,}6)\cos(x/2)$ | 1 | 2 |
| 7 | $2,6x^2\ln x$ | 1,2 | 2,2 |
| 8 | $(x^2+1)\sin(x\text{-}0{,}5)$ | 0,5 | 1,5 |
| 9 | $x^2\cos(x/4)$ | 2 | 3 |
| 10 | $\frac{\sin(0,2x-3)}{x^2+1}$ | 3 | 4 |
| 11 | $3x+\ln x$ | 1 | 2 |
| 12 | $4xe^{x^2}$ | -1 | 0 |

Рассчитать значение определенного интеграла можно с помощью формулы Ньютона-Лейбница

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

где $F(x)$ – первообразная подинтегральной функции.

В функциях интегрирования в Scilab реализованы различные численные алгоритмы. Наиболее универсальной командой интегрирования в Scilab является

*[I, err] = intg(a, b, name [,er1 [,er2]]),*

где *name* — имя функции, задающей подынтегральное выражение (функция может быть задана в виде набора дискретных точек, т.е. таблицей или с помощью внешней функции);

*a и b* – пределы интегрирования;

*erl и er2* — абсолютная и относительная точность вычислений (необязательные параметры).

Пример. Вычислить значение интеграла.

$$\int_2^5 \sin(x)\cdot\cos(2x)$$

Первообразной подинтегральной функции является функция

$$F(x) = -\underline{\cos(3x)/6} + \underline{\cos(x)/2}.$$

Решение, полученное по формуле Ньютона-Лейбница, приведено на рис. 22.

```
-->int=-cos(15)/6+cos(5)/2-(-cos(6)/6+cos(2)/2)
 int  =

    0.6365475
```

Рис. 22. Вычисление интеграла по формуле Ньютона-Лейбница

При использовании функции *intg* необходимо вначале задать подинтегральную функцию как внешнюю. Это можно сделать с помощью конструкции function … endfunction.

Решение, полученное с помощью функции *intg*, приведено на рис. 23.

```
-->function y=f(x),y=sin(x).*cos(2*x),endfunction;

-->[Int,er]=intg(2,5,f)
 er  =

    5.348D-14
 Int  =

    0.6365475
```

Рис. 23. Использование функции intg

На рис. 24 приведен файл-сценарий вычисления рассматриваемого интеграла.

интегралы.sci - Текстовый редактор

интегралы.sci

```
//Вычисление интеграла по формуле Ньютона-Лейбница
int=-cos(15)/6+cos(5)/2-(-cos(6)/6+cos(2)/2)
//
//Задание подинтегральной функции как внешней
function y=f(x),y=sin(x).*cos(2*x),endfunction;
//
//Вычисление интеграла функцией intg
[Int,er]=intg(2,5,f)
```

Рис. 24. Файл-сценарий вычисления рассматриваемого интеграла

## Аппроксимация экспериментальных данных

Очень часто, особенно при анализе эмпирических данных возникает необходимость найти в явном виде функциональную зависимость между величинами *x* и *y*, которые получены в результате измерений.

При аналитическом исследовании взаимосвязи между двумя величинами x и y производят ряд наблюдений и в результате получается таблица значений:

| $x$ | $x_1$ | $x_2$ | $\dots$ | $x_i$ | $\dots$ | $x_n$ |
|---|---|---|---|---|---|---|
| $y$ | $y_1$ | $y_2$ | $\dots$ | $y_i$ | $\dots$ | $y_n$ |

Эта таблица обычно получается как итог каких-либо экспериментов, в которых $x_i$ (независимая величина) задается экспериментатором, а $y_i$ получается в результате опыта. Поэтому эти значения $y_i$ будем называть эмпирическими или опытными значениями.

Между величинами x и y существует функциональная зависимость, но ее аналитический вид обычно неизвестен, поэтому возникает практически важная задача - найти эмпирическую формулу

$$y = f(x; a_1, a_2, \ldots, a_m),$$

(где $a_1, a_2, \ldots, a_m$ - параметры), значения которой при $x = x_i$ возможно мало отличались бы от опытных значений $y_i\ (i = 1,2,\ldots,n)$.

Метод наименьших квадратов позволяет по экспериментальным данным подобрать такую аналитическую функцию, которая проходит настолько близко к экспериментальным точкам, насколько это возможно.

Согласно методу наименьших квадратов наилучшими коэффициентами $a_1, a_2, \ldots, a_m$ считаются те, для которых сумма квадратов отклонений найденной эмпирической функции от заданных значений функции

$$S(a_1, a_2, \ldots, a_m) = \sum_{i=1}^{n} \left[ f(x_i; a_1, a_2, \ldots, a_m) - y_i \right]^2$$

будет минимальной.

Нахождение коэффициентов $a_i$ сводится к решению системы.

$$\begin{array}{l}
a_0 n + a_1 \Sigma\, x_i + a_2 \Sigma\, x_i^2 + \ldots + a_k \Sigma\, x_i^k = \Sigma\, y_i \\
a_0 \Sigma\, x_i + \quad a_1 \Sigma\, x_i^2 + a_2 \Sigma\, x_i^3 + \ldots + a_k \Sigma\, x_i^{k+1} = \Sigma\, x_i y_i \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_0 \Sigma\, x_i^k + \quad a_1 \Sigma\, x_i^{k+1} + a_2 \Sigma\, x_i^{k+2} + \ldots + a_k \Sigma\, x_i^{2k} = \Sigma\, x_i^k y_i
\end{array}$$

Построение эмпирической формулы состоит из двух этапов: выяснение общего вида этой формулы и определение ее наилучших параметров.

Эта система упрощается, если эмпирическая формула линейна относительно параметров $a_i$, тогда система - будет линейной.

Конкретный вид системы зависит от того, из какого класса эмпирических формул мы ищем зависимость. В случае линейной зависимости $y = a_1 + a_2 x$ система примет вид:

$$\begin{cases} a_1 n + a_2 \sum\limits_{i=1}^{n} x_i = \sum\limits_{i=1}^{n} y_i, \\ a_1 \sum\limits_{i=1}^{n} x_i + a_2 \sum\limits_{i=1}^{n} x_i^2 = \sum\limits_{i=1}^{n} x_i\, y_i. \end{cases}$$

Эта линейная система может быть решена любым известным методом (методом Гаусса, простых итераций, формулами Крамера).

Для реализации этой задачи в Scilab предусмотрена функция

$$[a,S] = datafit(F,z,c),$$

Где $F$ – аппроксимирующая функция, параметры которой необходимо подобрать;

$z$ – матрица исходных данных;

$c$ – вектор начальных приближений;

$a$ – вектор коэффициентов;

$S$ – сумма квадратов отклонений измеренных значений от расчетных

Вид аппроксимирующей функции, подбирается как наиболее подходящий для заданных экспериментальных данных, это может быть:

- Линейная,
- Логарифмическая,
- Полиномиальная,
- Экспоненциальная и др.

Чаще в качестве аппроксимирующей функции выбирают полином необходимой степени.

Пример. Пусть в результате эксперимента были получены некоторые данные, отображенные в виде таблицы. Требуется построить аналитическую зависимость, наиболее точно описывающую результаты эксперимента.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *x* | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 | 1.6 | 1.8 | 2.0 |
| *y* | 0.57 | 0.70 | 0.89 | 1.10 | 1.32 | 1.50 | 1.58 | 1.40 | 1.32 | 1.10 | 0.90 |

В качестве функции аппроксимирующей данные эксперимента следует взять полином третьей степени, у которого четыре коэффициента, которые необходимо найти.

На рис. 25 приведен файл-сценарий подбора аппроксимирующего полинома с помощью функции *datafit.*

*аппроксимация_2.sci

```
//Задание функции, вычисляющей разность между
//экспериментальными и теоретическими значениями
function [y]=P(c,z)
  y=z(2)-c(1)-c(2)*z(1)-c(3)*z(1)^2-c(4)*z(1)^3;
endfunction
//Задание исходных данных
x=[0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0];
y=[0.57 0.7 0.89 1.1 1.32 1.5 1.58 1.4 1.32 1.1 0.9];
//Построение графика экспериментальных данных
plot(x,y,'rx')
//
//Формирование матрицы исходных данных
z=[x;y];
//
//Формирование вектора начальных значений коэффициентов,размерность которого должна
//совпадать с количеством искомых коэффициентов
//искомых коэффициентов
c=[0;0;0;0];
//
//Определение коэффициентов полинома - a и суммы квадратов
//отклонений err
[a,err]=datafit(P,z,c)
//
//Задание вектора значений аргумента в диапазоне 0; 2
//Вычисление вектора значений найденного полинома для
//заданного вектора значений аргумента
t=[0:0.05:2];
pl=a(1)+a(2)*t+a(3)*t^2+a(4)*t^3;
//
//Построение графика найденного полинома
plot(t,pl)
xgrid()
```

Рис. 25. Файл-сценарий аппроксимации экспериментальных данных полиномом 3 степени

В результате работы функции *datafit* была подобрана аналитическая зависимость в виде полинома

$$P = -0{,}3x^3 + 0{,}17x^2 + 1{,}03x + 0{,}52\ ,$$

а сумма квадратов отклонений измеренных значений от расчетных составила 0,038 (рис. 26).

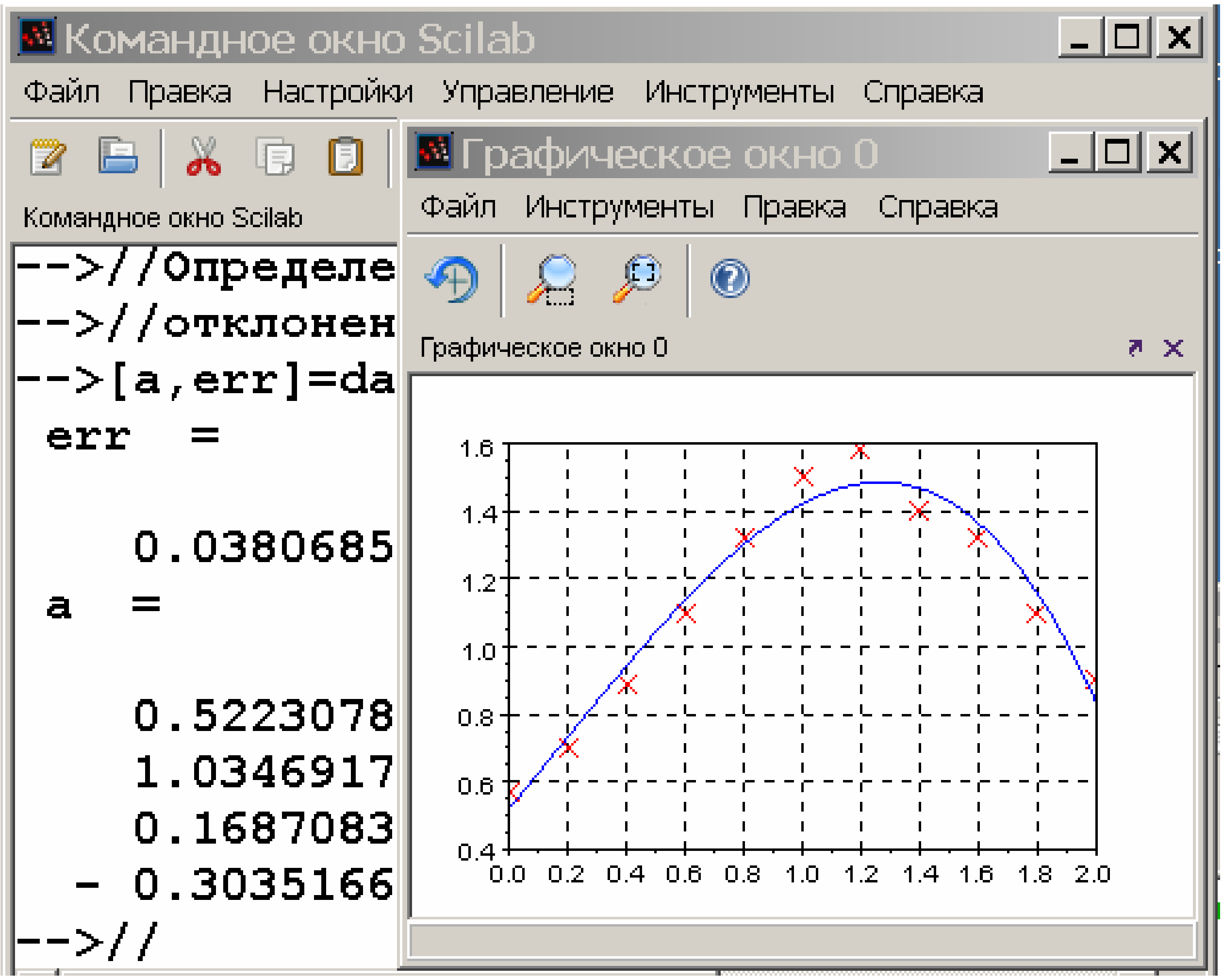



Рис. 26. Решение задачи аппроксимации

**Контрольные вопросы**

1. Квадратурные формулы прямоугольников (с оценкой точности).
2. Квадратурная формула трапеций (с оценкой точности).
3. Квадратурная формула Симпсона (с оценкой точности).
4. Квадратурные формулы гауссова типа.

## Лабораторная работа №4

Тема: **Решение обыкновенных дифференциальных уравнений.**

1. **Цель работы**

Использование методов решения обыкновенных дифференциальных уравнений для решения конкретных задач строительства.

2. **Учебные вопросы, подлежащие рассмотрению:**

Решение дифференциального уравнения:

- ✓ методом Коши,
- ✓ методом Эйлера,
- ✓ Методом Эйлера-Коши,
- ✓ Рунге-Кутта 4-го порядка
- ✓ Адамса.

3. **Порядок выполнения работы**

*Задание*

Решить задачу Коши для дифференциального уравнения *y=f(x,y)* на отрезке [*a*,*b*] при заданном начальном условии и шаге интегрирования *h*.

Номер варианта соответствует порядковому номеру в списке.

*Исполнение*: С помощью инструментальных пакетов MS Office методами Эйлера, Рунге-Кутта 4-го порядка и Адамса, предусмотрев вывод полученных решений в виде таблиц и графиков.

*Оценка*: Сопоставление полученных результатов, решаемых различными методами

Методические указания

Дана система дифференциальных уравнений:

$$\begin{cases} \dfrac{dy_1}{dx} = F_1(x, y_1, y_2, \dots, y_n) \\ \dfrac{dy_2}{dx} = F_2(x, y_1, y_2, \dots, y_n) \\ \dots \\ \dfrac{dy_n}{dx} = F_n(x, y_1, y_2, \dots, y_n) \end{cases}$$

, где n – *размерность системы.*

Рассмотрим задачу Коши для данной системы. Пусть известны начальные условия при $x_0 = a$: $y_1(x_0) = y_{10}$, $y_2(x_0) = y_{20}$, …, $y_n(x_0) = y_{n0}$. Требуется найти $y_1(x)$, $y_2(x)$,…, $y_n(x)$, проходящие через заданные точки: $(x_0,y_{10})$, $(x_0,y_{20})$, …, $(x_0,y_{n0})$.

Методы решения одного дифференциального уравнения можно обобщить и на их системы.

Метод Рунге-Кутта 4-го порядка для системы ОДУ 1-го порядка

Расчетные формулы метода Рунге-Кутта 4-го порядка для системы ОДУ 1-го порядка:

$$y_{i,j+1} = y_{i,j} + \frac{h}{6} \cdot (K_{oi} + 2K_{1i} + 2K_{2i} + K_{3i})$$

где $h = \dfrac{b-a}{m}$;

$a = x_0$;

$b = x_m$;

m – количество узлов;

$i = \overline{1,n}$ – номер функции;

$j = 0,1,\ldots,m-1$ – номер узла;

$K_{0i} = F_i(x_j, y_{1j}, y_{2j}, \ldots, y_{nj})$;

$K_{1i} = F_i(x_j + \frac{h}{2}, y_{1j} + \frac{h}{2} \cdot K_{0i}, y_{2j} + \frac{h}{2} \cdot K_{0i}, \ldots, y_{nj} + \frac{h}{2} \cdot K_{0i})$;

$K_{2i} = F_i(x_j + \frac{h}{2}, y_{1j} + \frac{h}{2} \cdot K_{1i}, y_{2j} + \frac{h}{2} \cdot K_{1i}, \ldots, y_{nj} + \frac{h}{2} \cdot K_{1i})$;

$K_{3i} = F_i(x_j + h, y_{1j} + hK_{2i}, y_{2j} + hK_{2i}, \ldots, y_{nj} + hK_{2i})$.

Для решения дифференциальных уравнений и систем в Sciab предусмотрена функция:

*[y,w,iw]=ode([type],y0,t0,t [,rtol [,atol]],f [,jac] [,w,iw])*

для которой, обязательными входными параметрами являются:

*y0* - вектор начальных условий;

*t0* - начальная точка интервала интегрирования;

*t* - координаты узлов сетки, в которых происходит поиск решения;

*f* - внешняя функция, определяющая правую часть уравнения или системы уравнений ;

*y* - вектор решений.

Для того чтобы решить обыкновенное дифференциальное уравнение вида

$$\frac{dy}{dt} = f(t, y),\ y(t_0) = y_0$$

необходимо вызвать функцию

*y=ode(y0,t0,t,f)* .

Рассмотрим необязательные параметры функции *ode*:

*type* - параметр с помощью которого можно выбрать метод решения или тип решаемой задачи, указав одну из строк:

– "adams" - применяют при решении дифференциальных уравнений или систем методом прогноза-коррекции Адамса;

– "stiff" - указывают при решении жестких задач;
– "rk" - используют при решении дифференциальных уравнений или систем методом Рунге_Кутта четвертого порядка;
– "rkf" - указывают при выборе пятиэтапного метода Рунге_Кутта четвертого порядка; "fix" - тот же метод Рунге_Кутта, но с фиксированным шагом;

*rtol,atol* - относительная и абсолютная погрешности вычислений, вектор, размерность которого совпадает с размерностью вектора *y*, по умолчанию *rtol=0.00001, atol=0.0000001* , при использовании параметров "*rkf*" и "*fix*" - *rtol=0.001, atol=0.0001*;

*jac* - матрица, представляющая собой якобиан правой части жесткой системы дифференциальных уравнений, задают матрицу в виде внешней функции вида *J=jak(t,y)*;

*w,iw* - векторы, предназначенные для сохранения информации о параметрах интегрирования, которые применяют для того, чтобы последующие вычисления выполнялись с теми же параметрами.

Рассмотрим использование функции на примере следующих задач.

ЗАДАЧА 8.1. Решить задачу Коши

$$\frac{dx}{dt}+x=\sin(xt),\ x(0)=1.5.$$

Перепишем уравнение следующим образом:

$$\frac{dx}{dt}=-x+\sin(x\,t),\ x(0)=1.5.$$

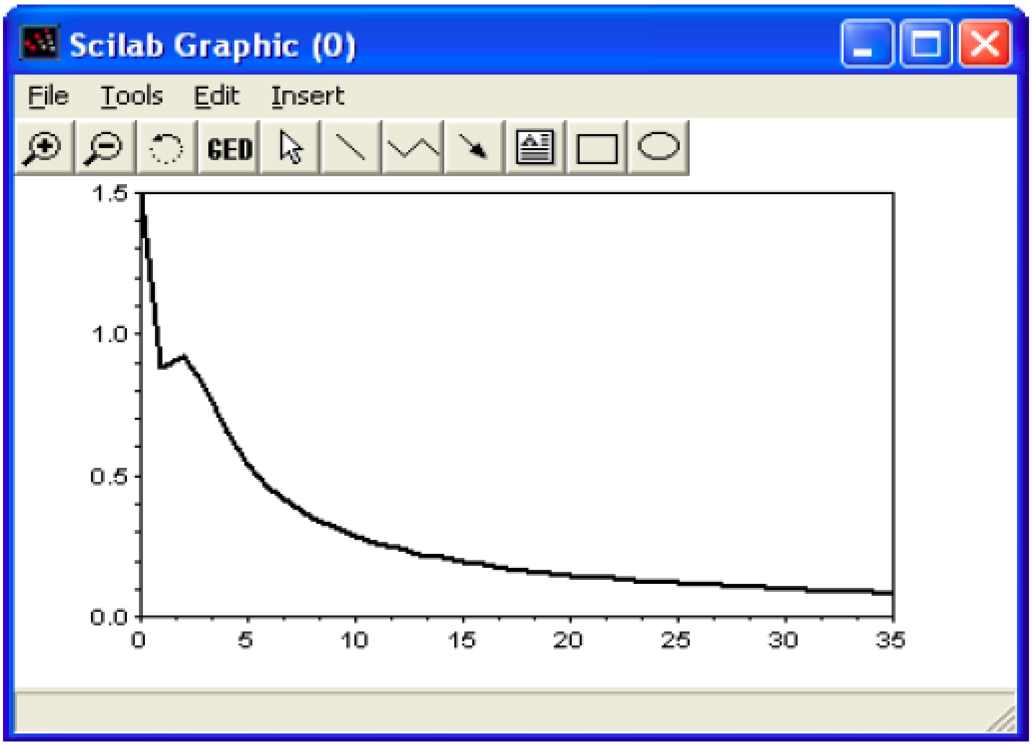
Далее представим его в виде внешней функции, так как показано в листинге и применим функцию *y=ode(x0,t0,t,f),* в качестве параметров которой будем использовать

*f* - ссылка на предварительно созданную функцию *f(t,x)*;
*t* - координаты сетки;
*x0,t0* - начальное условие *x(0)=1.5*;
*y* - результат работы функции.

График, моделирующий процесс, описанный заданным уравнением, представлен на рис.8.1.

```
-->function yd=f(t,x),yd=-x+sin(t*x),endfunction;
-->x0=1.5;t0=0;t=0:1:35;
-->y=ode(x0,t0,t,f);
-->plot(t,y)
```

Листинг 8.1




из задачи 8.1

ЗАДАЧА 8.2. Решить задачу Коши

$$x' = \cos(x\,y),$$
$$y' = \sin(x + t\,y),$$
$$x(0) = 0, y(0) = 0.$$

на интервале [0; 10].

Листинг 8.2 содержит функцию, описывающую заданную систему обыкновенных дифференциальных уравнений и команды Sciab необходимые для ее численного и графического решения (рис.8.2).

```
>>%Функция, описывающая систему дифференциальных уравнений
-->function dy=syst(t,y)
-->dy=zeros(2,1);
-->dy(1)=cos(y(1)*y(2));
-->dy(2)=sin(y(1)+y(2)*t);
-->endfunction
>>%Решение системы дифференциальных уравнений
-->x0=[0;0];t0=0;t=0:1:10;
-->y=ode(x0,t0,t,syst);
>>%Формирование графического решения
-->plot(t,y)
```

Листинг 8.2

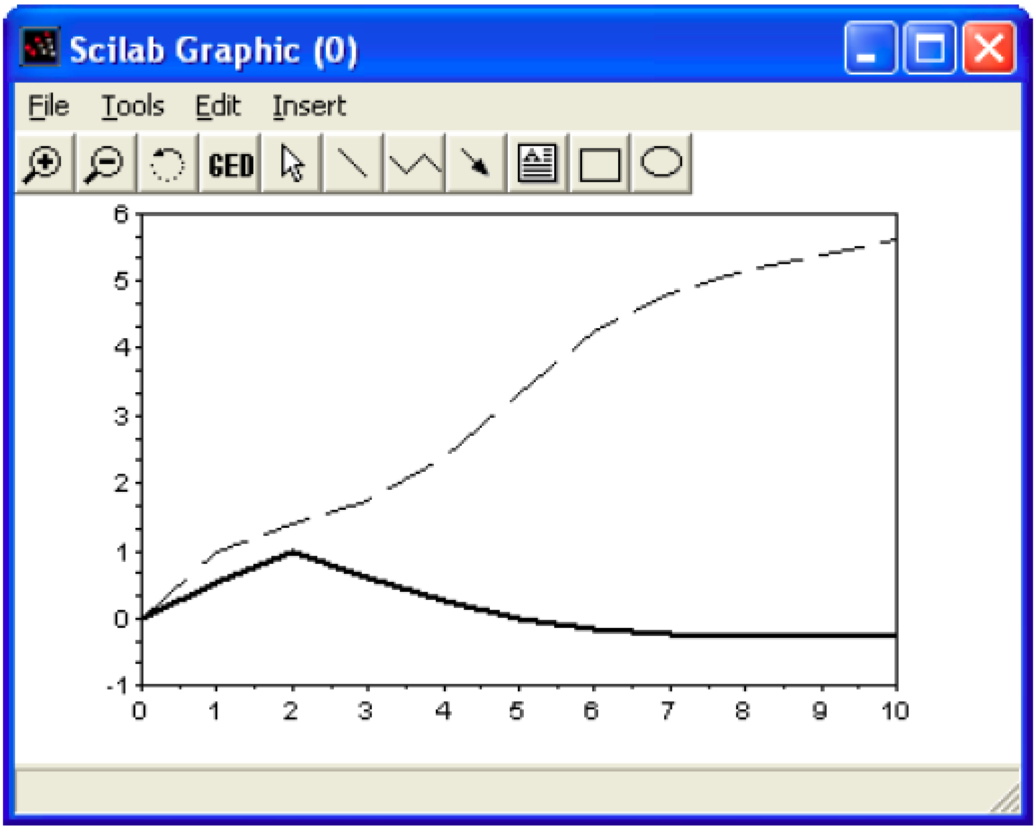



Рис. 8.2. Решение задачи 8.2

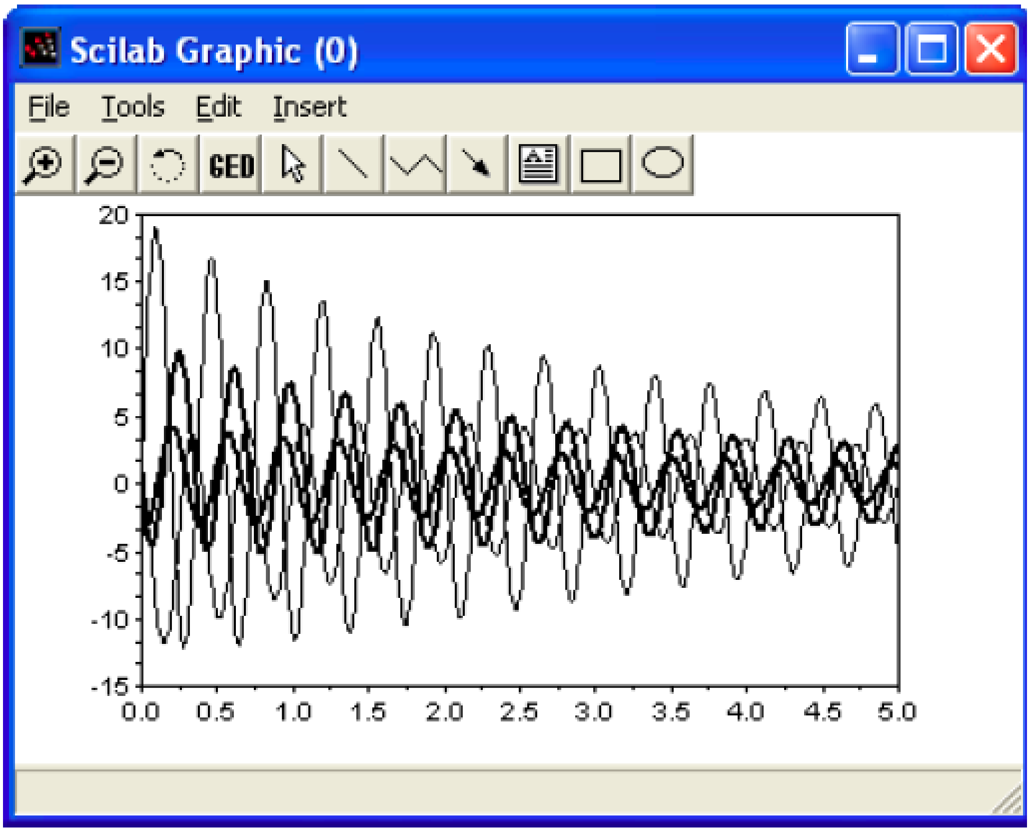



Рис. 8.3. Графическое решение жесткой системы

ЗАДАЧА 8.3. Найти решение задачи Коши для следующей жесткой системы:

$$\frac{dX}{dt}=\begin{pmatrix}119.46 & 185.38 & 126.88 & 121.03\\ -10.395 & -10.136 & -3.636 & 8.577\\ -53.302 & -85.932 & -63.182 & 54.211\\ -115.58 & -181.75 & -112.8 & -199\end{pmatrix}X\,;\,X(0)=\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

Решение системы показано в листинге 8.3.
Графическое решение показано на рис. 8.3.

```
-->B=[119.46 185.38 126.88 121.03;-10.395 -10.136 -3.636 8.577;
-->-53.302 -85.932 -63.182 -54.211;-115.58 -181.75 -112.8 -199];
-->function dx=syst1(t,x), dx=B*x,endfunction
-->function J=Jac(t,y),J=B,endfunction
-->x0=[1;1;1;1]; t0=0; t=0:0.01:5;
-->y=ode("stiff",x0,t0,t,syst1,Jacobian);
-->plot(t,y)
```

Листинг 8.3

ЗАДАЧА 8.5. Решить следующую краевую задачу

$$\frac{d^2x}{dt^2}+4\frac{dx}{dt}+13=e^{\sin(t)},\ x(0.25)=-1,\ x'(0.25)=1.$$

на интервале [0.25; 2].
Преобразуем уравнение в систему, сделав замену:

$$y=\frac{dx}{dt}$$

$$\frac{dy}{dt}=-4\text{y}-13\text{x}+e^{\sin(t)},\ \frac{dx}{dt}=y,\ y(0.25)=1,\ x(0.25)=-1.$$

Составим функцию вычисления системы и решим ее так, как показано в листинге 8.5. График решения приведен на рис. 8. 5.

```
function F=FF(t,x)
F=[-4*x(1)-13*x(2)+exp(t);x(1)];
endfunction
-->//Решение системы дифференциальных уравнений
-->X0=[1;-1];t0=0.25;t=0.25:0.05:2;
-->y=ode("stiff",X0,t0,t,FF);
-->//Вывод графика решения
-->plot(t,y)
```

Листинг 8.5

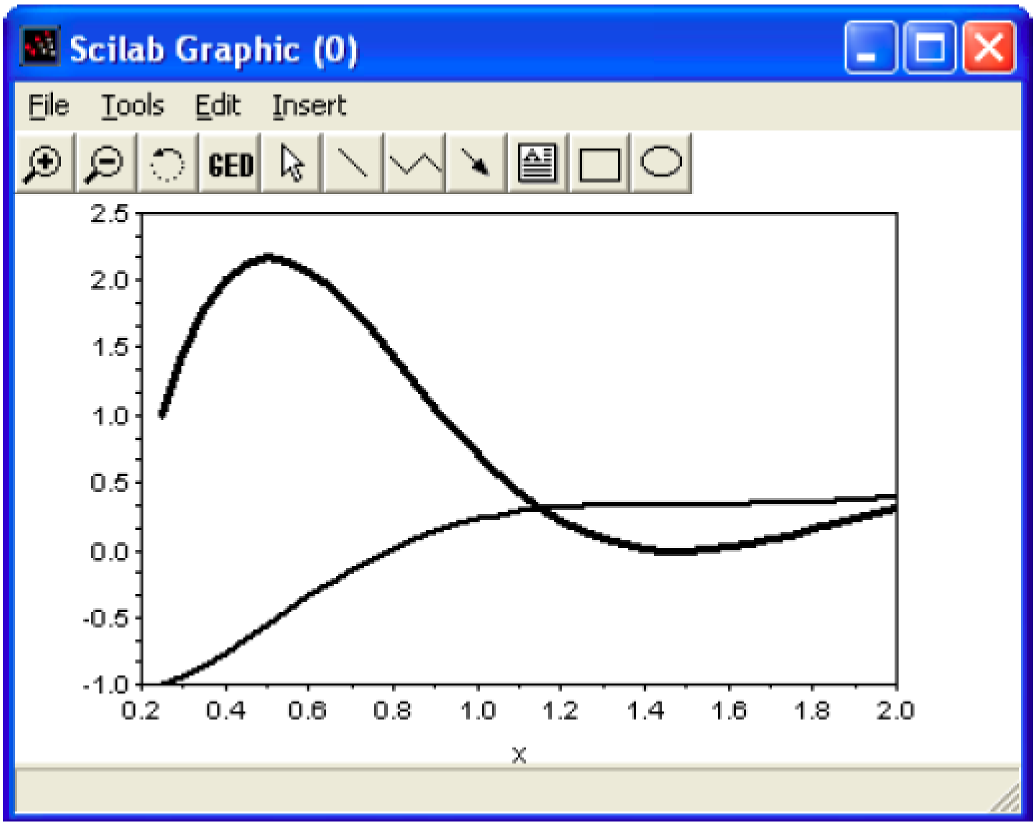



Рис.8.5. Решение задачи 8.5

**Контрольные вопросы**

1) Сформулируйте задачу Коши для обыкновенных дифференциальных уравнений первого порядка.

2) Что является решением дифференциального уравнения: а) в высшей математике, б) в прикладной математике?

3) Какие методы решения дифференциальных уравнений называются одношаговыми, многошаговыми? Приведите примеры.

4) Сравните решения, полученные на первом, втором шаге методами Эйлера, Рунге-Кутта и разложением в ряд Тейлора (трудоемкость, погрешность…).

5) Как оценить погрешность применяемого метода? Как ее уменьшить?

6) Сравните одношаговые и многошаговые методы решения дифференциальных уравнений, указав достоинства и недостатки первых и вторых.

7) Что такое экстраполяционные и интерполяционные методы (формулы) Адамса?

8) Можно ли применять: а) только экстраполяционные методы Адамса, б) только интерполяционные?

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В каком виде представляются все данные в Scilab ?
2. Как вводятся элементы вектора-строки ?
3. Как обратиться к блоку последовательно расположенных элементов вектора?
4. Какие знаки используются для поэлементного умножения, деления, возведения в степень векторов?