Лабораторная работа №1. Знакомство с пакетом Scilab.

Цель работы

Научиться использовать пакет **Scilab** для типовых вычислений теории автоматического управления.

Задачи работы

1. Освоить основные способы ввода числовых и символьных данных, а также вычислений математических выражений.
2. Изучить средства построения графиков системы **Scilab**.

Краткие теоретические сведения

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;
- задачи 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.

Несмотря на то, что система Scilab содержит достаточное количество встроенных команд, операторов и функций, отличительная ее черта — это гибкость. Пользователь может создать любую новую команду или функцию, а затем использовать ее наравне со встроенными. К тому же, система имеет достаточно мощный собственный язык программирования высокого уровня, что говорит о возможности решения новых задач.

Пользователи, знакомые с системой MATLAB (MathWorks Inc.), обнаружат практически полное сходство в правилах работы и синтаксисе команд с системой Scilab. Это действительно так. По этой причине можно использовать самоучители и справочники по MATLAB для самостоятельного изучения Scilab. Отличие Scilab в том, что она распространяется бесплатно.

Среда Scilab

После запуска Scilab на экране появиться *основное окно приложения*. Окно содержит *меню*, *панель инструментов* и *рабочую область*. Признаком того, что

система готова к выполнению команды, является наличие приглашения “-->”, после которого расположен активный (мигающий) курсор. Рабочую область со знаком приглашения называют *консоль*. *Ввод команд* в Scilab осуществляется с клавиатуры. Нажатие клавиши Enter заставляет систему выполнить команду и вывести результат (рис. 1).

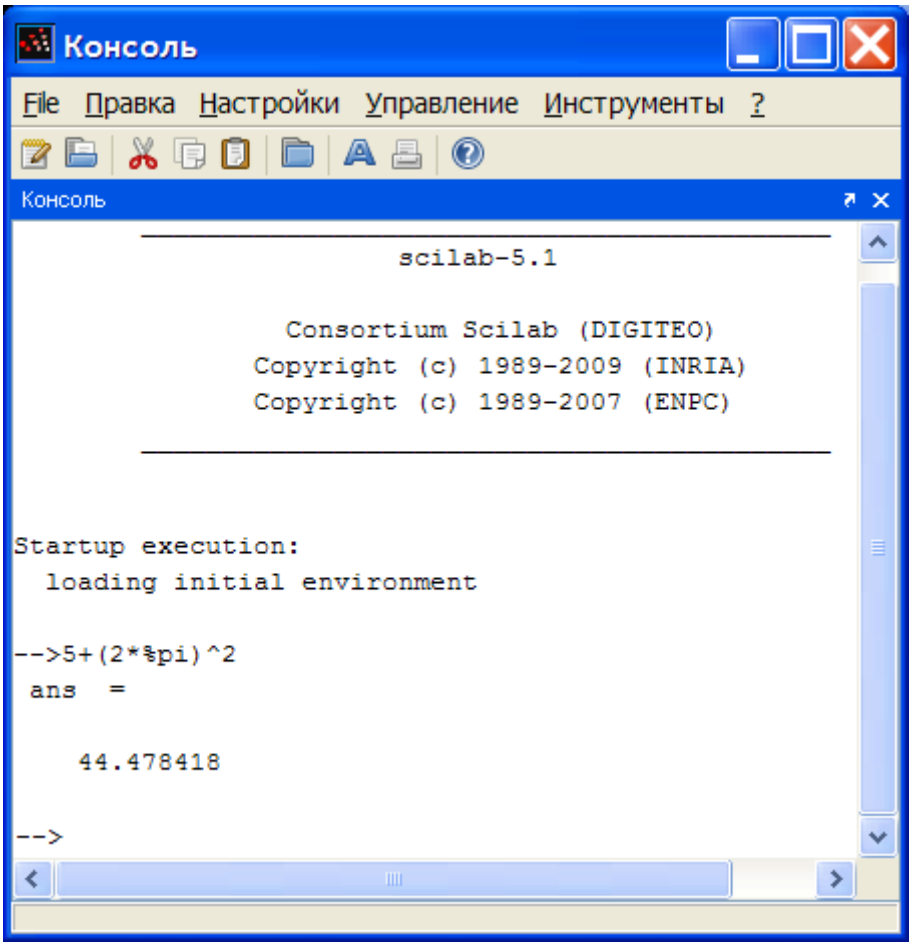

Рис. 1 Выполнение элементарной команды в Scilab

Понятно, что все выполняемые команды не могут одновременно находиться в поле зрения пользователя. Просмотреть ту информацию, которая покинула видимую часть окна можно, если воспользоваться полосами прокрутки или колесиком скроллинга «мыши».

Клавиши «Стрелка вверх» и «Стрелка вниз» в консоли Scilab имеют особое назначение. Эти клавиши позволяют вернуть в командную строку ранее введенные команды, так как все они сохраняется в специальной области памяти — *журнале команд*. Так, если в пустой активной командной строке нажать клавишу ↑, то появится последняя вводимая команда, повторное нажатие ↑ вызовет предпоследнюю и так далее. Клавиша ↓ выводит команды в обратном порядке. Таким образом, можно сказать, что вся информация в рабочей области находится или в *зоне просмотра* или в *зоне редактирования*.

Важно знать, что в *зоне просмотра* нельзя ничего исправить или ввести. Единственная допустимая операция, кроме просмотра, это выделение информации с

помощью мыши и копирование ее в буфер обмена, например, для дальнейшего помещения в командную строку. *Зона редактирования* — это фактически командная строка. В ней действуют элементарные приемы редактирования: → — перемещение курсора вправо на один символ; ← — перемещение курсора влево на один символ; Home — перемещение курсора в начало строки; End — перемещение курсора в конец строки; Del — удаление символа после курсора; Backspace — удаление символа перед курсором.

Кроме того, существуют особенности *ввода команд*. Если команда заканчивается точкой с запятой «;», то результат ее действия не отображается в командной строке. В противном случае, при отсутствии знака «;», результат действия команды сразу же выводится в рабочую область (листинг 1).

```
-->%pi
 %pi  =

    3.1415927

-->a=2*%pi;

-->a
 a  =

    6.2831853
-->
```

Листинг 1

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*. При желании, определения всех переменных и функций, входящих в текущую сессию (т.н. *окружение*), можно сохранить в виде файла, саму сессию сохранить нельзя.

Все команды, завершенные нажатием клавиши Enter, сохраняются в *журнале команд*. Журнал команд не очищается автоматически при завершении работы с системой. Для очистки журнала используется специальная команда меню.

Основные команды главного меню

В данном разделе рассмотрим команды, отражающие некоторые специфические особенности Scilab версии 5.1, и с которыми нам придется часто иметь дело.

File

Набор подпунктов меню **File** типичен для приложений Windows и интуитивно понятен. Следует обратить внимание на подпункты **Сменить текущий каталог…** и **Отобразить текущий каталог**. Русская локализация пакета выполнена не вполне корректно: названия каталогов с кириллицей (кодировка Windows-1251) отражаются нечитаемо в консоли Scilab (кодировка Юникод Utf-8). Возникают проблемы и при чтении файлов, в полном имени которых присутствуют символы кириллицы. Текущий каталог по умолчанию — тот, из которого произведен запуск приложения. Если использовалась иконка на рабочем столе, то текущим каталогом будет «Рабочий стол» в профиле текущего пользователя. Такое имя использовать нельзя. Поэтому после запуска Scilab первым делом выполняем команды [меню — **File — Сменить текущий каталог…**] и указываем, например, C:\Scilab\Work (папка должна существовать). Другой вариант — создать ярлык для запуска Scilab в том каталоге, в котором предполагается работать.

Правка

Объяснения подпунктов этого меню не требуется.

Настройки

Здесь можно поэкспериментировать с цветами и шрифтами консоли (хотя установки по умолчанию вполне рациональны). Здесь же можно очистить *журнал команд* (см. выше) или окно консоли.

Управление

Команды **продолжить**, **остановить**, **прервать** используются для управления объемными и трудоемкими вычислениями.

Инструменты

- **Редактор** — вызов сеанса редактора программ (сценариев). Того же эффекта можно достичь командой “editor”, набранной в консоли и подтвержденной клавишей Enter.

Замечание. Из-за несовершенства локализации, после вызова редактора, сразу измените кодировку на Utf-8 (по крайней мере, если планируется вывод на консоль кириллических символов). Для этого откройте пункт **options** меню редактора, выберите подпункт **encoding**, и отметьте птичкой **Utf-8**.

{Казалось бы, проще изменить кодовую страницу консоли, однако, автору этих строк это до сих пор не удалось…}

- **Scicos** — запуск графического редактора для конструирования и симуляции моделей гибридных динамических систем. Это средство также может быть запущено из консоли командой “scicos”.

Scicos — аналог MATLAB Simulink, с меньшими возможностями и несколько иными приемами работы. Приверженцам MATLAB это средство визуального моделирования может показаться «тугим и неповоротливым» но, перефразируя известную пословицу, «бесплатному коню зубы не смотрят!». К тому же, кон-

сорциум Scilab–INRIA–ENPC не стоит на месте: комфортность и популярность пакета будут возрастать. Большинство лабораторных работ данного цикла выполняются с применением именно этого средства.

Текстовые комментарии

Текстовый комментарий в Scilab — это строка, начинающаяся с символов //. Использовать текстовые комментарии можно как в рабочей области, так и в тексте файла-сценария. Строка после символов // не воспринимается как команда и нажатие клавиши Enter приводит к активизации следующей командной строки:

```
-->//2+8
-->
```

Листинг 2

Элементарные математические выражения

Для выполнения простейших *арифметических операций* в Scilab применяют следующие операторы:

+ сложение,
– вычитание,
* умножение,
/ деление слева направо,
\ деление справа налево,
^ возведение в степень.

Вычислить значение арифметического выражения можно, если ввести его в командную строку и нажать клавишу ENTER. В рабочей области появится результат:

```
-->1.5*(2-1.78/3.14)^2
 ans  =

    3.0807538
-->
```

Листинг 3

Если вычисляемое выражение *слишком длинное*, то перед нажатием клавиши ENTER следует набрать три или более точек. Это будет означать продолжение командной строки:

```
-->1+2+3+4+5+6....
-->+7+8+9+10+....
-->11+12+13+14+15
 ans  =
```

```
    120.
-->
```

Листинг 4

Переменные в Scilab

В рабочей области Scilab можно определять *переменные*, а затем использовать их в выражениях. Любая переменная до использования в формулах и выражениях должна быть определена. Для *определения переменной* необходимо набрать имя переменной, символ «=» и значение переменной. Здесь знак равенства — это *оператор присваивания*, действие которого не отличается от аналогичных операторов языков программирования.

Имя переменной не должно совпадать с именами встроенных процедур, функций и встроенных переменных системы и может содержать до 24 символов. Система различает большие и малые буквы в именах переменных. То есть `SUM`, `Sum`, `sum` — это имена разных переменных. Выражение в правой части оператора присваивания может быть числом, арифметическим выражением, строкой символов или символьным выражением. Если речь идет о символьной или строковой переменной, то выражение в правой части оператора присваивания следует брать в одинарные или двойные кавычки.

```
-->a="Мама мыла ";

-->b='раму';

-->c=a+b+'!'
 c  =

 Мама мыла раму!
-->
```

Листинг 5

Для очистки значения переменной и освобождения рабочей памяти можно применить команду

```
-->clear имя_переменной;
```

Системные переменные Scilab

Если команда не содержит знака присваивания, то по умолчанию вычисленное значение присваивается специальной *системной переменной* `ans`. Причем полученное значение можно использовать в последующих вычислениях, но важно помнить, что значение `ans` изменяется после каждого вызова команды без оператора присваивания.

Другие *системные переменные* в Scilab начинаются с символа `%`:

- `%i` — мнимая единица ($\sqrt{-1}$)

- `%pi` — число π = 3.141592653589793;
- `%e` — число e = 2.7182818;
- `%inf` — машинный символ бесконечности (∞);
- `%nan` — неопределенный результат (0/0, ∞/∞ и т.п.);
- `%eps` — условный ноль %eps = 2.220D–16 ($2.22 \cdot 10^{-16}$).

Функции в Scilab

Все *функции*, используемые в Scilab, можно разделить на два класса:

- встроенные;
- определенные пользователем.

В общем виде *обращение к функции* в Scilab имеет вид:

```
имя_переменной = имя_функции(переменная1 [,переменная2,…])
```

где

`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;

`переменная1, переменная2, …` — список аргументов функции.

Пакет Scilab снабжен достаточным количеством всевозможных встроенных функций, знакомство с которыми будет происходить по мере необходимости. Здесь приведем только элементарные математические функции, используемые чаще всего.

`sin(x)` — синус числа *x*
`cos(x)` — косинус числа *x*
`tan(x)` — тангенс числа *x*
`cotg(x)` — котангенс числа *x*
`asin(x)` — арксинус числа *x*
`acos(x)` — арккосинус числа *x*
`atan(x)` — арктангенс числа *x*
`exp(x)` — экспонента числа *x*
`log(x)` — натуральный логарифм числа *x*
`log10(x)` — десятичный логарифм от числа *x*
`log2(x)` — логарифм по основанию два от числа *x*
`sqrt(x)` — корень квадратный из числа *x*
`abs(x)` — модуль числа *x*

Рассмотрим несколько способов создания *функций пользователя* в Scilab.

Первый способ — это применение оператора `deff`, который в общем виде можно записать так:

```
deff('[s1,s2,...]=newfunction(e1,e2,....)',text [,opt])
```

где

`s1, s2, …` — список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений,

`e1, e2, …` — входные параметры,

`text` — собственно тело функции, представляющее собой одно или последовательность математических выражений в виде текстовых строк,

`opt` — дополнительный параметр, используется для «тонкой» настройки профиля.

Например, функция, вычисляющая корни квадратного уравнения $ax^2+bx+c=0$ по формулам $x_1=\frac{-b+\sqrt{D}}{2a}$, $x_2=\frac{-b-\sqrt{D}}{2a}$, где $D=b^2-4ac$, может быть определена так:

```
-->deff('[x1,x2]=korni(a,b,c)',['D=b^2-4*a*c';...
-->                             'x1=(-b+sqrt(D))/(2*a)';...
-->                             'x2=(-b-sqrt(D))/(2*a)'])
-->//Обращение к функции korni
-->[x1,x2]=korni(1,2,2)
 x2  =

  - 1. - i
 x1  =

  - 1. + i

-->
```

Листинг 6

Второй способ создания функции — это применение конструкции вида:

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

где

`имя1,...,имяN` — список выходных параметров, т.е. переменных, которым будет присвоен конечный результат вычислений (параметров может быть от 1 до N),

`имя_функции` — имя, с которым эта функция будет вызываться,

`переменная_1,...,переменная_M` — входные параметры (параметров может быть от 1 до M).

`тело функции` — последовательность математических выражений, записанных по правилам Scilab.

Все имена переменных внутри функции, а так же имена из списка входных и выходных параметров воспринимаются системой как *локальные*, то есть эти переменные считаются определенными только внутри функции.

Вообще говоря, функции в Scilab играют роль подпрограмм. Поэтому целесообразно набирать их тексты в редакторе и сохранять в виде отдельных файлов. Причем имя файла должно совпадать с именем функции. Расширение файлам-функциям обычно присваивают .sce (сценарий) или .sci (скайлаб-файл).

Обращение к функции осуществляется так же, как и к любой другой встроенной функции системы, то есть из командной строки. Однако функции, хранящиеся в отдельных файлах, должны быть предварительно загружены в систему, например при помощи оператора `exec(имя_файла)` или командой (меню — **File — Выполнить...)**, что в общем, одно и то же.
Замечание. Команда `exec(имя_файла)` требует явного указания расширения.

Массивы и матрицы в Scilab

Для работы с множеством данных удобно использовать массивы. В этом случае, вместо создания переменной для хранения каждого данного, достаточно создать один массив, где каждому элементу будет присвоен порядковый номер. Таким образом, *массив* — множественный тип данных, состоящий из фиксированного числа элементов. Как и любой другой переменной, массиву должно быть присвоено имя.

Переменную, представляющую собой просто список данных, называют *одномерным массивом* или *вектором*. Для доступа к данным, хранящимся в определенном элементе массива, необходимо указать *имя массива* и *порядковый номер* этого элемента, называемый *индексом*.

Для хранения данных в виде таблиц, в формате строк и столбцов, необходимо использовать *двумерные массивы* (матрицы). Для доступа к данным, хранящимся в таком массиве, необходимо указать имя массива и два индекса. Первый должен соответствовать номеру строки, а второй — номеру столбца, на пересечении которых хранится необходимый элемент.

Значение *нижней границы индексации* в Scilab равно единице. Индексы могут быть только целыми положительными числами.

Самый простой способ задать одномерный массив в Scilab имеет вид:

```
[name]=Xn:dX:Xk
```

где
`name` — имя переменной, в которую будет записан сформированный массив,
`Xn` — значение первого элемента массива,
`Xk` — значение последнего элемента массива,
`dX` — шаг, с помощью которого формируется каждый следующий элемент массива, то есть значение второго элемента составит Xn+dX, третьего Xn+ dX+dX и так далее до Xk. Если параметр dX в конструкции отсутствует, то принимается значение по умолчанию, равное единице.

Переменную заданную как массив можно использовать в арифметических выражениях и в качестве аргумента математических функций. Результатом работы таких операторов являются массивы.

Еще один способ задания векторов и матриц в Scilab — это их *поэлементный ввод*. Так, для определения *вектор-строки* следует ввести имя массива, а затем после знака присваивания, в квадратных скобках через пробел или запятую перечислить элементы массива:

```
V_str=[0 1 4 9 16 25]
```

или

```
Day=['Пн','Вт','Ср','Чт','Пт','Сб','Вс']
```

Элементы *вектора-столбца* вводятся через точку с запятой:

```
X=[10; 0; -5]
```

При поэлементном вводе матриц используется комбинация этих способов: элементы строк вводятся через пробелы или запятые, а строки разделяются точкой с запятой:

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

 A  =

    1.    2.    3.
  - 1.    2.  - 3.
    0.    1.    2.
```

Обратиться к элементу массива можно, указав имя массива и индексы элемента в круглых скобках: `Day(3)` или `A(2,2)`.

Особое значение при обращении к элементам массива играет символ двоеточия «:», который следует интерпретировать как «любой индекс» или «все индексы». Так, запись `A(:,3)` означает «все элементы 3-го столбца», т.е. третий столбец матрицы A, а запись `A(1,:)` — первая строка массива A.

Для записи матричных выражений в Scilab используются следующие операторы:

- + сложение;
- − вычитание;
- ' транспонирование;
- * умножение;
- ^ возведение в степень
- \ левое деление;
- / правое деление;
- .* поэлементное умножение;
- .^ поэлементное возведение в степень;
- .\ поэлементное левое деление;
- ./ поэлементное правое деление.

Примечание. Левое деление `A\B` означает $\mathbf{A}^{-1}\mathbf{B}$, а правое `A/B` означает $\mathbf{A}\mathbf{B}^{-1}$.

В Scilab имеется множество функций для всевозможных матричных операций. С некоторыми из них мы познакомимся в процессе выполнения лабораторных работ. В основном же рекомендуется справочная литература. Ниже приведен краткий список полезных функций.

`matrix(A[,n,m])` преобразует матрицу A в матрицу другого размера;
`ones(m,n)` создает матрицу единиц из `m` строк и `n` столбцов;
`zeros(m,n)` создает нулевую матрицу из `m` строк и `n` столбцов;
`eye(m,n)` формирует единичную матрицу из `m` строк и `n` столбцов;
`diag(V[,k])` возвращает квадратную матрицу с элементами `V` на главной диагонали или на `k`-й ; функция `diag(A[,k])`, где `A` — ранее определенная матрица, в качестве результата выдаст вектор-столбец, содержащий элементы главной или `k`-ой диагонали матрицы `A`;
`cat(n,A,B,[C,…])` объединяет матрицы `A` и `B` или все входящие матрицы, при `n`=1 по строкам, при `n`=2 по столбцам; то же что [A; B] или [A, B];
`size(V[,fl])` определяет размер массива `V`, если `V` двумерный массив, то `size(V,1)` или `size(V,'r')` определяют число строк матрицы `V`, а `size(V,2)` или `size(V,'c')` — число столбцов;
`det(M)` вычисляет определитель квадратной матрицы `M`;
`rank(M[,tol])` вычисление ранга матрицы `M` с точностью `tol`;
`spec(M)` вычисляет собственные значения и собственные векторы квадратной матрицы `M`.
`inv(A)` вычисляет матрицу, обратную `A`;
`linsolve(A,b)` решает систему линейных алгебраических уравнений вида `Ax+b=0`;

`rref(A)` осуществляет приведение матрицы `A` к треугольной форме, используя метод исключения Гаусса;
`lu(M)` выполняет треугольное разложение матрицы `M` (LU-разложение);

Построение двумерных графиков

Графическое окно

Графические приложения в пакете Scilab выводятся в отдельном окне, называемом графическим. Любая функция для построения и вывода графика создаст (если оно еще не было создано) графическое окно. Одновременно может быть открыто несколько графических окон, которым присваиваются порядковые номера (начиная с 0). Однако вывод графической информации может осуществляться только в одно окно, называемое текущим. Создать графическое окно и (или) выбрать текущее из созданных, можно командой `scf(номер_окна)`. Команда `scf()` без параметров создаст новое графическое окно с номером, на единицу большим максимального из открытых в текущий момент, или с номером 0, если ни одного графического окна еще не было создано. Заголовок текущего графического окна можно изменить с помощью команды `xname("Заголовок окна")`.

Для очистки графического окна с заданным номером и установки его текущим служит функция `xbasc(Номер_окна)`. Команда `xbasc()` без параметров очищает текущее графическое окно. Команда `xbasc(1:4)` очистит графические окна с первого по четвертое, а `xbasc([1,3,5])` — первое, третье и пятое. Любопытно то, что этой же командой можно создать пустое графическое окно (или окна) с заданными номерами: прежде чем очистить окно, оно будет создано. В отличие от нее, функция `clf(номер_окна)` не будет создавать окно перед очисткой.

Чтобы «убить» графическое окно, т.е. удалить с экрана и освободить его номер, используется команда `xdel(номер_окна)` (или щелчок мыши на красном крестике в правом верхнем углу графического окна).

Функция plot()

Для построения одного, или нескольких графиков в одной системе координат, можно обратиться к функции `plot` следующим образом:

```
plot(x1,y1,x2,y2,… xn,yn)
```

где
`x1,y1` — массивы абсцисс и ординат первого графика;
`x2,y2` — массивы абсцисс и ординат второго графика;

...

Пример. Построить на одном графике зависимости $y_1 = \sin(x_1)$ на интервале $x_1 \in [-4\pi; 4\pi]$ и $x_2 = \mathrm{tg}(y_2)$ на интервале $y_2 \in [-1{,}5; 1{,}5]$.

```
-->//Задаем вектор значений x1 с шагом 0.01

-->x1=-4*%pi: 0.01: 4*%pi;

-->y1=sin(x1);

-->//Задаем вектор значений y2 с шагом 0.01

-->y2=-1.5: 0.01: 1.5;

-->x2=tan(y2);

-->plot(x1,y1,x2,y2);

-->xtitle("Тригонометрические функции","Ось X","Значения");

-->xgrid; //Показать линии сетки

-->
```

Листинг 7

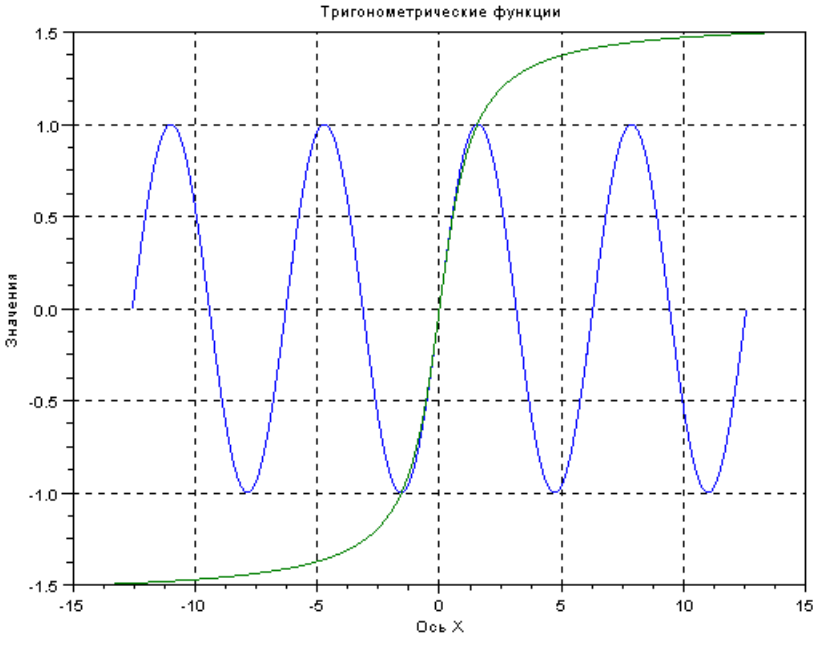

Рис. 2 Графическое окно к листингу 7

Несомненное достоинство графических окон Scilab — легкий и корректный их перенос в другие приложения Windows. Для этого в графическом окне вызыва-

ем команду меню **File — Экспортировать…** или **File — Копировать в буфер обмена**.

Функция plot2d()

Следующей функцией, которая может быть использована для построения двумерных графиков, является функция **plot2d**. Существуют аналогичные функции более высокого уровня: **plot2d1** эквивалентна команде **plot2d**. **plot2d2**: аналогична команде **plot2d**, но выглядит как гистограмма без вертикальных делений на прямоугольники (огибающая гистограммы). Является кусочно-непрерывной функцией. **plot2d3**: аналогична команде **plot2d**, но кривая изображается в виде столбчатой диаграммы, у которой прорисованы только вертикальные линии. **plot2d4**: аналогична команде **plot2d**, но кривая изображается стрелочками.

Синтаксис функции **plot2d**

plot2d([x],y)
plot2d([x],y,[opt_args])

Параметры **x**,**y**: две матрицы или два вектора.

Если **y** является вектором, то **x** должен быть вектором той же размерности. Если значение **x** не задано, оно полагается равным вектору [1:]. Это означает, что если `y=[2.2,4.4,9.3]`, то команда `plot2d(y)` будет считать, что `x=[1,2,3]`. *Замечание*: Применяя такие конструкции, очень легко получить в дальнейшем ошибку.

Если **y** является матрицей, то **x** может быть:

1) вектором размера, равного размерности столбца матрицы **y**. Каждый столбец матрицы **y** будет изображаться относительно вектора **x**.
2) матрицей того же размера, что и **y**. Каждый столбец **y** изображается относительно соответствующего столбца **x**.
3) если **x** не задан, он полагается равным вектору [1:[row dimension of y]].

[opt_args]: указывает последовательность соответствующих значений ключей **key1=value1**, **key2=value2**,... где **key1**, **key2**,... могут принимать одно из следующих значений:

style: устанавливает стиль для каждой кривой. Принимаемые значения смотри ниже.

rect: устанавливает диапазоны оцифровки осей графика. Присваиваемое значение должно быть вещественным вектором размера 4 [xmin, ymin, xmax, ymax].

logflag: устанавливает тип шкалы по осям (линейная – n или логарифмическая – l). Допускается присвоение следующих строковых значений: `"nn"`, `"nl"`, `"ln"` и `"ll"`.

frameflag: управляет вычислением фактических координатных диапазонов от минимальных требуемых значений. Присоединенное значение должно быть целым числом в пределах от от 0 до 8 (см. ниже).

axesflag: устанавливает, как будут нарисованы оси и границы графика. Принимает целые значения от 0 до 5. Пояснения ниже.

nax: устанавливает число основных и промежуточных засечек осей. Присоединенное значение должно быть действительным вектором с четырьмя целочисленными элементами [nx, Nx, ny, Ny]. Здесь Nx (Ny) — число основных делений (с подписями) на оси X (Y); nx (ny) — число промежуточных делений.

leg: устанавливает заголовок (подпись) кривой. Присоединенное значение должно быть строковой константой. Если кривых несколько, используется конструкция `"leg1@leg2@...."`.

Перечисленные выше ключи могут принимать следующие значения:

style: этот параметр может использоваться, чтобы определить, как кривые нарисованы. Присоединенное значение должно быть вектором размером, равным числу кривых.

Если **style(i)** строго положителен, кривая отображается как простая линия, и значение **style(i)** определяет её цвет (см. getcolor). Отметим, что тип линии и её толщина могут быть установлены посредством свойств объекта ломаной линии (см. polyline_properties). Между заданными точками кривой выполняется кусочно-линейная интерполяция.

Если **style(i)** меньше или равен нулю, точки кривой отображаются метками, вид которых определяется модулем **style(i)**. Отметим, что цвет меток и их размеры могут быть установлены посредством свойств объекта ломаной линии (см. polyline_properties).

style(i)	**Описание метки**	**style(i)**	**Описание метки**
0	Точка	-8	Плюс, вписанный в ромб
-1	Плюс	-9	Кружок
-2	Крестик	-10	Звездочка
-3	Плюс, вписанный в окружность	-11	Квадрат
-4	Закрашенный ромб	-12	Треугольник верш. вправо
-5	Незакрашенный ромб	-13	Треугольник верш. влево
-6	Треугольник вершиной вверх	-14	Пятиконечная звезда
-7	Треугольник вершиной вниз		

frameflag: управляет вычислением фактических координатных диапазонов от минимально требуемых. Фактические диапазоны могут быть больше, чем минимально требуемые.

frameflag=0: вычисления не производятся. График использует предыдущий масштаб (или масштаб по умолчанию).

frameflag=1: фактический диапазон — диапазон, заданный параметром rect.

frameflag=2: фактический диапазон вычислен из минимального/максимального значения данных x и y.

frameflag=3: используется диапазон, заданный параметром rect и увеличенный, чтобы получить изометрический масштаб.

frameflag=4: фактический диапазон вычислен из минимального/максимального значения данных x и y и увеличен, чтобы получить изометрический масштаб.

frameflag=5: используется диапазон, заданный параметром rect и увеличенный, чтобы получились красивые обозначения осей.

frameflag=6: фактический диапазон вычислен из минимального/максимального значения данных x и y и увеличен, чтобы получились красивые обозначения осей.

frameflag=7: фактический диапазон — диапазон, заданный параметром rect. Предыдущий график (графики) перестраивается в новом масштабе. Используется для добавления графика к уже построенным.

frameflag=8: аналогично **frameflag=2**, но предыдущий график (графики) перестраивается в новом масштабе.

frameflag=9: аналогично **frameflag=8**, но диапазон увеличивается, чтобы получить красивые обозначения осей. Это значение параметра по умолчанию.

axesflag: управляет отрисовкой осей.

axesflag	Границы графика (box)	Ось X	Ось Y	Положение оси Y
0	нет	нет	нет	—
1	есть	есть	есть	слева
2	есть	нет	нет	—
3	нет	есть	есть	справа
4	нет	есть	есть	через (0,0)
5	есть	есть	есть	через (0,0)
9(по умолчанию)	нет	есть	есть	слева

Символьные выражения и вычисления

Большой интерес представляют возможности символьных вычислений в системе Scilab. Основной структурной единицей символьных вычислений является полином, определяемый функцией **poly()**.

Функция poly()

Синтаксис:

p = poly(a, vname, ["flag"])

a — вещественное число или матрица;

vname — символьное имя переменной в полиноме: строка длиной до четырех символов (если задать имя длиннее, лишние символы не будут учтены).

"flag" — строка, имеющая два возможных значения: "roots", "coeff" (возможны сокращения 'r' или 'c'). В первом случае матрица **a** задает корни полинома, а во втором — непосредственно его коэффициенты (начиная с младшего). Значение по умолчанию – "roots".

```
-->//Определить полином второй степени с коэффициентами 1,2,3
-->a=[1 2 3];
-->p=poly(a,"x","coeff")
p  =

                2
    1 + 2x + 3x
-->//Определить полином, имеющий корни -1, +1:
-->a=[-1 1];
-->q=poly(a,"x")
q  =

          2
  - 1 + x
-->//Определить символьную переменную x:
-->x=poly(0,"x")
x  =

    x
-->//Выполнить символьные вычисления:
-->w=x*p/q
w  =

           2    3
    x + 2x + 3x
    -----------
             2
     - 1 + x
```

```
-->//Найти корни полинома числителя:
-->roots(numer(w))
ans  =

    0
  - 0.3333333 + 0.4714045i
  - 0.3333333 - 0.4714045i
-->//Найти корни полинома знаменателя:
-->roots(denom(w))
ans  =

    1.
  - 1.
```

Листинг 8

Символьные матрицы

Из полиномов могут быть составлены матрицы, с которыми, в свою очередь, возможно выполнение матричных операций.

```
-->x=poly(0,"x");
--> H=[x, x*x+2; 1-x, 1+x]
 H  =

                   2
    x         2 + x

    1 - x     1 + x
-->//Обращение символьной матрицы:
--> H1=invr(H)
 H1  =

                          2
      1 + x         - 2 - x
    ---------       ---------
               3                3
  - 2 + 3x + x    - 2 + 3x + x

    - 1 + x             x
    ---------       ---------
               3                3
  - 2 + 3x + x    - 2 + 3x + x
-->//Вычисление определителя символьной матрицы:
-->d=detr(H)
d  =

               3
  - 2 + 3x + x
```

Листинг 9

Порядок выполнения работы

1. Начало работы. Подготовка рабочего каталога. Запуск Scilab

1.1. Создать на диске средствами операционной системы рабочий каталог C:\Scilab\Work. Если каталог уже существует, очистить его.

1.2. Скопировать с рабочего стола в каталог, созданный на предыдущем шаге, ярлык Scilab. (Именно *скопировать*, а не переместить!)

Примечание. Если этого ярлыка на рабочем столе нет, то создайте ярлык средствами Windows, указав путь к файлу:
"C:\Program Files\scilab-5.1\bin\WScilex.exe"

1.3. В рабочем каталоге (C:\Scilab\Work) выполнить двойной клик на ярлыке Scilab. Раскроется окно *консоли*, как на рис. 1.

1.4. Выполнить команду [меню — **File — Отобразить текущий каталог**]. Убедиться в правильности отображения имени каталога.

1.5. Выполнить команду [меню — **File — Сменить текущий каталог…**]. В раскрывшемся окне проводника выбрать каталог с кириллическим именем (например, …\Мои документы\Мои рисунки). Убедиться в невозможности выполнения команды.

Примечание. Данный пункт демонстрирует особенность версии Scilab-5.1. Возможно, Ваш экземпляр программного обеспечения лишен этого недостатка. Тогда сделанные выше замечания о символах кириллицы нужно игнорировать.

2. Работа в режиме диалога

2.1. Набрать в командной строке (в строке с приглашением -->) «%pi» (без кавычек) и нажать Enter. В дальнейшем все команды, набранные в командной строке, завершаем нажатием клавиши Enter. Как по-вашему, что означают символы «%pi», «%e», «%i»?

2.2. Выполнить команды, приведенные на листингах 3 – 9.

2.3. Найти корни полиномов (самостоятельно, по вариантам):

№ вар-та	Полиномы для решения
1	$1.1x^4 - x - 0.9 = 0$ $x^3 + x - 4 = 0$
2	$2x^4 - x - 1.5 = 0$ $3x^3 - 5x^2 + 9x - 10 = 0$
3	$2x^4 - 9.25x^2 - 63x + 5 = 0$ $3x^3 - 21x + 2 = 0$
4	$0.9x^4 + 4.2x^3 - 8.5x^2 - 13 = 0$ $5x^3 + 13x - 11 = 0$
5	$3x^4 + 4x^3 - 12x^2 - 5 = 0$ $x^3 + 2x^2 + 2 = 0$
6	$3.2x^4 + 7.75x^3 + 6.3x^2 - 10.5 = 0$ $2x^3 + 0.48x^2 + 1.6x - 2.6 = 0$
7	$2x^4 - 3x^2 - 5 = 0$ $2x^3 - 0.35x^2 + 0.85x + 1 = 0$
8	$1.05x^4 - 17x^2 + 6 = 0$ $2x^3 - 0.35x^2 + 0.85x + 1 = 0$
9	$3.25x^4 + 7.67x^3 + 5x^2 - 11 = 0$ $2x^3 + 5x^2 + 11x + 7 = 0$
10	$2.2x^4 - 1.2x^2 - 11 = 0$ $3x^3 - 0.42x^2 + 0.95x - 2 = 0$
11	$-x^4 - 18x^2 + 6 = 0$ $2x^3 - 0.08x^2 + 0.94x + 1.3 = 0$
12	$-1.21x^4 + x^3 + 2x^2 - 3x - 5 = 0$ $3x^3 - 13x^2 + 16x - 15 = 0$
13	$0.89x^4 + 3.67x^3 - 7.92x^2 - 13 = 0$ $2x^3 - 0.35x^2 + 0.47x - 1.43 = 0$
14	$6x^4 + 8x^3 - 23x^2 + 2.1 = 0$ $5x^3 + 20x^2 + 5x + 8 = 0$

Копию рабочего окна с решением задачи поместить в отчет.

3. Создание сценариев

3.1. Написать сценарий, выполняющий решение системы линейных алгебраических уравнений:

№ вар-та	Система	№ вар-та	Система
1	$\begin{cases} -x_1 - x_2 - 2x_3 - 3x_4 = 2 \\ 3x_1 - x_2 - x_3 - 2x_4 = -8 \\ 2x_1 + 3x_2 - x_3 - x_4 = -12 \\ x_1 + 2x_2 + 3x_3 - x_4 = 8 \end{cases}$	2	$\begin{cases} x_1 - 2x_2 + 3x_3 - 2x_4 = -6 \\ x_1 + x_2 - 2x_3 - 3x_4 = -8 \\ 3x_1 - 2x_2 - x_3 + 2x_4 = 4 \\ 2x_1 + 3x_2 + 2x_3 + x_4 = 8 \end{cases}$
3	$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = 5 \\ 2x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ 3x_1 + 2x_2 + x_3 + 2x_4 = 1 \\ 4x_1 + 3x_2 + 2x_3 + x_4 = -5 \end{cases}$	4	$\begin{cases} 0.1x_1 + 0.5x_2 + 0.3x_3 - 0.4x_4 = 2 \\ 0.3x_1 + 0.1x_2 - 0.2x_3 = 0.9 \\ 0.5x_1 - 0.7x_2 + x_4 = -0.9 \\ 0.3x_2 - 0.5x_3 = 0.1 \end{cases}$
5	$\begin{cases} 10x_2 + 30x_3 + 40x_4 = -50 \\ 10x_1 + 20x_3 + 30x_4 = -40 \\ 30x_1 + 20x_2 - 50x_4 = 120 \\ 40x_1 + 30x_2 + 50x_3 = 50 \end{cases}$	6	$\begin{cases} 0.3x_1 + x_2 + 1.67x_3 - 2.3x_4 = 4 \\ 3x_1 + 5x_2 + 7x_3 - x_4 = 0 \\ 5x_1 + 7x_2 + x_3 - 3x_4 = 4 \\ 7x_1 + x_2 + 3x_3 - 5x_4 = 16 \end{cases}$
7	$\begin{cases} 2x_1 + x_2 + 5x_3 + x_4 = 8 \\ 0.333x_1 - x_2 - 2x_4 = 3 \\ 2x_2 + x_3 + 2x_4 = -5 \\ x_1 + 4x_2 + 7x_3 + 6x_4 = 0 \end{cases}$	8	$\begin{cases} -x_1 + x_2 + x_3 + x_4 = 12 \\ 2x_1 + x_2 + 2x_3 + 3x_4 = 13 \\ 1.5x_1 + x_2 + 0.5x_3 + x_4 = 7 \\ 4x_1 + 3x_2 + 2x_3 + x_4 = -15 \end{cases}$
9	$\begin{cases} -2x_1 - x_2 + 3x_3 + 2x_4 = 40 \\ -x_1 + x_2 + x_3 + 0.6667x_4 = 20 \\ -3x_1 - x_2 - x_3 + 2x_4 = 60 \\ -3x_1 - x_2 + 3x_3 - x_4 = 60 \end{cases}$	10	$\begin{cases} 3x_1 - 6x_2 - 3x_3 + 3x_4 = 8 \\ 2x_1 - x_2 + x_3 + x_4 = 5 \\ x_1 + x_2 + 2x_3 + x_4 = -1 \\ x_1 - x_2 - x_3 + 3x_4 = 10 \end{cases}$
11	$\begin{cases} 20x_1 + 5x_2 + 5x_4 = -9 \\ x_1 - 3x_2 + 4x_3 = -7 \\ 3x_2 - 2x_3 - 4x_4 = 12 \\ x_1 + 2x_2 - x_3 + 3x_4 = 10 \end{cases}$	12	$\begin{cases} x_1 - 3x_2 + x_3 + x_4 = 11 \\ x_1 + 3x_2 + 5x_3 + 7x_4 = 12 \\ 3x_1 + 5x_2 + 7x_3 + x_4 = 0 \\ -5x_1 - 7x_2 - x_3 - 3x_4 = -4 \end{cases}$
13	$\begin{cases} 2x_1 + x_2 + x_3 - x_4 = 11 \\ 2x_1 + x_2 - 3x_4 = 2 \\ 3x_1 + x_3 + x_4 = -3 \\ 4x_1 - 4x_2 - 4x_3 + 10x_4 = 7 \end{cases}$	14	$\begin{cases} 2x_1 + x_3 + 4x_4 = 19 \\ x_1 + 2x_2 - x_3 + x_4 = 18 \\ 2x_1 + x_2 + x_3 + x_4 = 15 \\ 2x_1 - 2x_2 + 4x_3 + 2x_4 = -11 \end{cases}$

В сценарии предусмотреть проверку. Листинг сценария и диалог рабочего окна при его запуске занести в отчет.

3.2. Написать сценарий для вычисления матрицы, обратной матрице D (если это возможно):

№	D	№	D
1	$\mathbf{D}=2(\mathbf{A}^2+\mathbf{B})(2\mathbf{B}-\mathbf{A})$, где $\mathbf{A}=\begin{bmatrix}2&3&-1\\4&5&2\\-1&0&7\end{bmatrix},\mathbf{B}=\begin{bmatrix}-1&0&5\\0&1&3\\2&-2&4\end{bmatrix}$	2	$\mathbf{D}=3\mathbf{A}-(\mathbf{A}+2\mathbf{B})\mathbf{B}^2$, где $\mathbf{A}=\begin{bmatrix}4&5&-2\\3&-1&0\\4&2&7\end{bmatrix},\mathbf{B}=\begin{bmatrix}2&1&-1\\0&1&3\\5&7&3\end{bmatrix}$
3	$\mathbf{D}=3\mathbf{A}^2-(\mathbf{A}+2\mathbf{B})\mathbf{B}$, где $\mathbf{A}=\begin{bmatrix}4&5&-2\\3&-1&0\\4&2&7\end{bmatrix},\mathbf{B}=\begin{bmatrix}2&1&-1\\0&1&3\\5&7&3\end{bmatrix}$	4	$\mathbf{D}=(\mathbf{A}-2\mathbf{B}^2)(2\mathbf{A}+\mathbf{B}^3)$, где $\mathbf{A}=\begin{bmatrix}5&2&0\\10&4&1\\7&3&2\end{bmatrix},\mathbf{B}=\begin{bmatrix}3&6&-1\\-1&-2&0\\2&1&3\end{bmatrix}$
5	$\mathbf{D}=2(\mathbf{A}-\mathbf{B})(\mathbf{A}^2+\mathbf{B})$, где $\mathbf{A}=\begin{bmatrix}5&1&7\\-10&-2&1\\0&1&2\end{bmatrix},\mathbf{B}=\begin{bmatrix}2&4&1\\3&1&0\\7&2&1\end{bmatrix}$	6	$\mathbf{D}=(\mathbf{A}-\mathbf{B})^2\mathbf{A}+2\mathbf{B}$, где $\mathbf{A}=\begin{bmatrix}5&-1&3\\0&2&-1\\-2&-1&0\end{bmatrix},\mathbf{B}=\begin{bmatrix}3&7&-2\\1&1&-2\\0&1&3\end{bmatrix}$
7	$\mathbf{D}=(\mathbf{A}^2-\mathbf{B}^2)(\mathbf{A}+\mathbf{B}^2)$, где $\mathbf{A}=\begin{bmatrix}7&2&0\\-7&-2&1\\1&1&1\end{bmatrix},\mathbf{B}=\begin{bmatrix}0&2&3\\1&0&-2\\3&1&1\end{bmatrix}$	8	$\mathbf{D}=2\mathbf{A}-(\mathbf{A}^2+\mathbf{B})\mathbf{B}$, где $\mathbf{A}=\begin{bmatrix}1&4&2\\2&1&-2\\0&1&-1\end{bmatrix},\mathbf{B}=\begin{bmatrix}4&6&-2\\4&10&1\\2&4&-5\end{bmatrix}$
9	$\mathbf{D}=2(\mathbf{A}-0.5\mathbf{B})+\mathbf{A}^3\mathbf{B}$, где $\mathbf{A}=\begin{bmatrix}5&3&-1\\2&0&4\\3&5&-1\end{bmatrix},\mathbf{B}=\begin{bmatrix}1&4&16\\-3&-2&0\\5&7&2\end{bmatrix}$	10	$\mathbf{D}=(\mathbf{A}-\mathbf{B})\mathbf{A}^2+3\mathbf{B}$, где $\mathbf{A}=\begin{bmatrix}3&2&-5\\4&2&0\\1&1&2\end{bmatrix},\mathbf{B}=\begin{bmatrix}-1&2&4\\0&3&2\\-1&-3&4\end{bmatrix}$
11	$\mathbf{D}=3(\mathbf{A}^2+\mathbf{B}^2)-2\mathbf{AB}$, где $\mathbf{A}=\begin{bmatrix}4&2&1\\3&-2&0\\0&-1&2\end{bmatrix},\mathbf{B}=\begin{bmatrix}2&0&2\\5&-7&-2\\1&0&-1\end{bmatrix}$	12	$\mathbf{D}=2\mathbf{A}^3+3\mathbf{B}(\mathbf{AB}-2\mathbf{A})$, где $\mathbf{A}=\begin{bmatrix}1&-1&0\\2&0&-1\\1&1&1\end{bmatrix},\mathbf{B}=\begin{bmatrix}5&3&1\\-1&2&0\\-3&0&0\end{bmatrix}$
13	$\mathbf{D}=\mathbf{A}(\mathbf{A}^2-\mathbf{B})-2(\mathbf{B}+\mathbf{A})\mathbf{B}$, где $\mathbf{A}=\begin{bmatrix}2&3&1\\-1&2&4\\5&3&0\end{bmatrix},\mathbf{B}=\begin{bmatrix}2&7&13\\-1&0&5\\5&13&21\end{bmatrix}$	14	$\mathbf{D}=(2\mathbf{A}-\mathbf{B})(3\mathbf{A}+\mathbf{B})-2\mathbf{A}^2\mathbf{B}$, где $\mathbf{A}=\begin{bmatrix}1&0&3\\-2&0&1\\-1&3&1\end{bmatrix},\mathbf{B}=\begin{bmatrix}7&5&2\\0&1&2\\-3&-1&-1\end{bmatrix}$

Листинг сценария и диалог рабочего окна при его запуске занести в отчет.

3.3. Написать сценарий для отображения в графическом окне графика функции $f(x)$.

№ вар.	$f(x)$	№ вар.	$f(x)$
1	$f(x)=\dfrac{1.2x^3+x^2-2.8x-1}{x^2-1}$	2	$f(x)=\dfrac{1.9x^3-2.8x^2-1.9x+1}{3x^2-1}$
3	$f(x)=\dfrac{2x^2-5}{\sqrt{x^2-2}}$	4	$f(x)=\dfrac{4.1x^3-3.25x}{4x^4-1}$
5	$f(x)=\dfrac{x^2-11.5}{4x-3}$	6	$f(x)=\dfrac{2.3x^2-7}{\sqrt{3x^2-4}}$
7	$f(x)=\sqrt[3]{(x-4.5)^2(x+2)}$	8	$f(x)=\sqrt[3]{x^2(x-4.7)}$
9	$f(x)=\sqrt[3]{(x+5)^2}-\sqrt[3]{(x-7)^2}$	10	$f(x)=\sqrt[3]{(x^2-x-2)^2}$
11	$f(x)=\sqrt[3]{x^2(x+3.5)^2}$	12	$f(x)=\sqrt[3]{(x+5)^2}-\sqrt[3]{x-1}$
13	$f(x)=\sqrt[3]{(4+x)(x^2+2x+1)}$	14	$f(x)=\sqrt[3]{(x^2-x-6)^2}$

Листинг сценария и копию графического окна занести в отчет.

3.4. Написать сценарий для отображения в графическом окне графика функции, заданной в полярных координатах:

№ вар-та	Функция	№ вар-та	Функция
1	$r(\varphi)=-2\operatorname{ctg}\varphi$	2	$r(\varphi)=2\cos 6\varphi$
3	$r(\varphi)=2^{\varphi}+1$	4	$r(\varphi)=2\sqrt{\cos 2\varphi}$
5	$r(\varphi)=3\varphi+2$	6	$r(\varphi)=3\varphi^2+\varphi$
7	$r(\varphi)=2\sin 6\varphi$	8	$r(\varphi)=3^{\varphi}$
9	$r(\varphi)=2\operatorname{tg}3\varphi$	10	$r(\varphi)=\dfrac{1}{\cos\dfrac{3}{\varphi}}$
11	$r(\varphi)=\dfrac{2}{\sin\varphi}+3$	12	$r(\varphi)=5\sin^2\dfrac{\varphi}{3}$
13	$r(\varphi)=5\sin\dfrac{\varphi}{3}$	14	$r(\varphi)=\dfrac{3}{\varphi^2}+1$

Листинг сценария и копию графического окна занести в отчет.