

Лабораторная работа: Система MATLAB для научных и инженерных расчетов.

Цели работы : Ознакомление с принципами использования системы MATLAB для научных и инженерных расчетов, выполнение расчетных работ с использованием матричных операций и специальных функций системы, моделирование.

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ РАБОТЫ**

Основным типом объектов системы MATLAB является прямоугольная матрица, элементами которой могут быть как вещественные так и комплексные числа. Скалярным переменным соответствует матрица размерности  $[1 \times 1]$ , а векторам - матрица с одним столбцом или строкой.

Матрица - это таблица данных, но особенность заключается в том, что для матриц установлены определенные правила выполнения математических операций. В системе MATLAB предусмотрены возможности выполнения матричных операций (например умножение матриц), а также специальные операции поэлементной обработки матриц как простых таблиц или массивов (Array) чисел. Для указания на режим поэлементной обработки операндов при выполнении математической операции перед символом операции ставится дополнительная точка (примеры см. ниже).

Система MATLAB является средой - интерпретатором команд, рабочее поле экрана используется для ввода данных и операторов.

В системе MATLAB применяется также способ вычисления с использованием программ, написанных пользователем. Программой является внешний M - файл (текстовый файл с расширением .m), который может создаваться и модифицироваться любым текстовым редактором, но с обязательным условием экспорта в виде ASCII файла (без специальных символов редактирования и разметки документа). Выполнение программы начинается при наборе имени M-файла программы в командной строке рабочего поля системы.

Результаты работы системы при выполнении команд или M- файла программы выводятся на рабочее поле экрана и на графики, для которых открываются специальные окна.

При запуске системы открывается командное окно рис.1, содержащее позиции меню:

**File Edit Debug Desktop Window Help** (Version 7), панель инструментов (линейку кнопок команд), указатель рабочего каталога и две строки с напоминанием о порядке вызова справочной системы и демонстрационных примеров.

Двойная стрелка » указывает на начало командной строки.



Рис.1

На экране можно открыть несколько окон системы. По команде меню **Desktop > Desktop Layout > Default** будут открыты окна **Current Directory** (с закладкой на второе окно **Workspace**) и **Command History**, рис.2.

Назначение кнопок панели инструментов представлено на рис.3.

На рис.2 показан вид рабочего окна системы MatLab (Version 7 (R14)). В левой части экрана находится окно Workspace и Command History. В окне Editor открыты два m-файла (Prim\_001.m и Prim\_002.m), результаты работы программы выведен на график Figure 1.

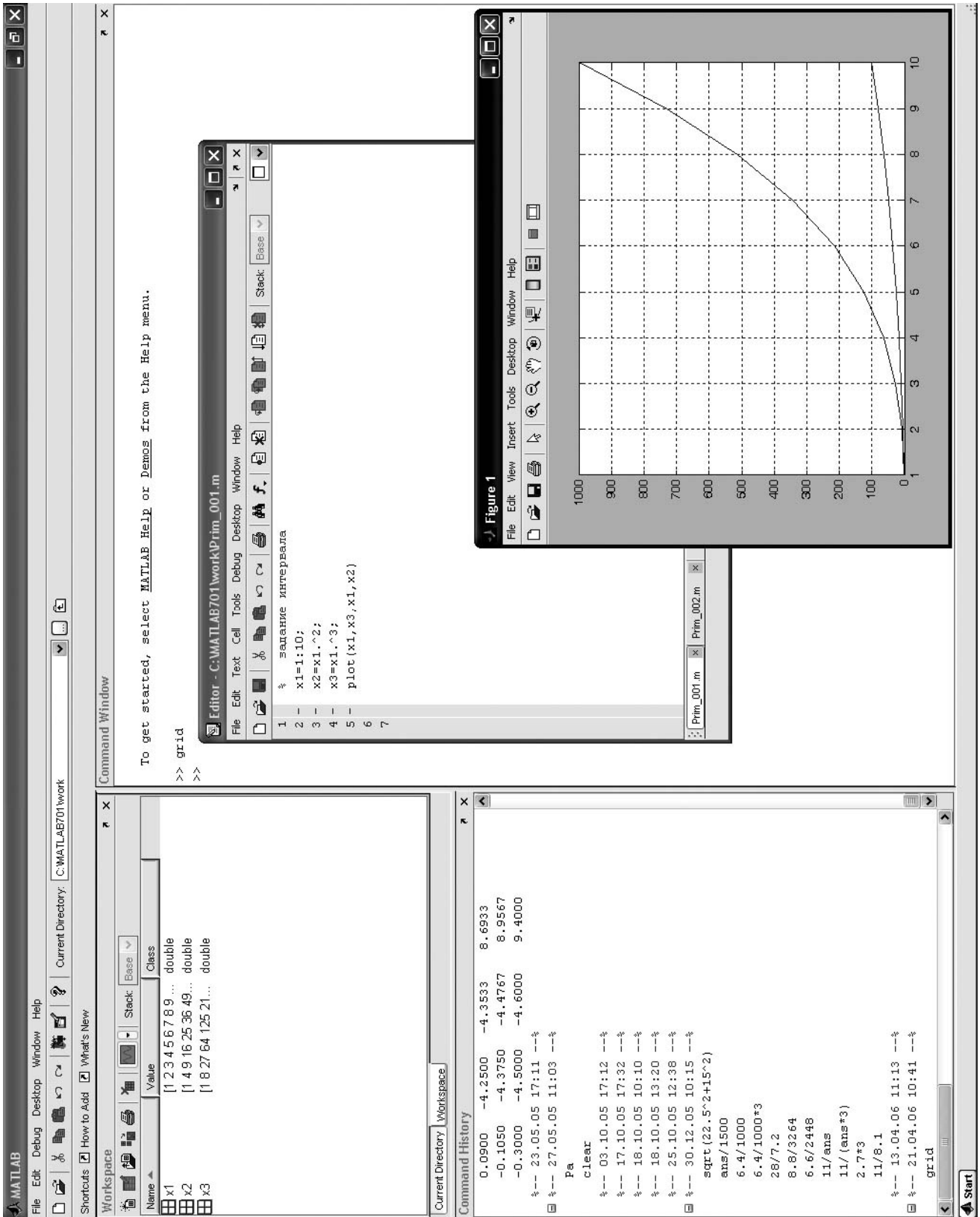


Рис.2.  
Рабочее окно системы MatLab (Version 7 (R14))

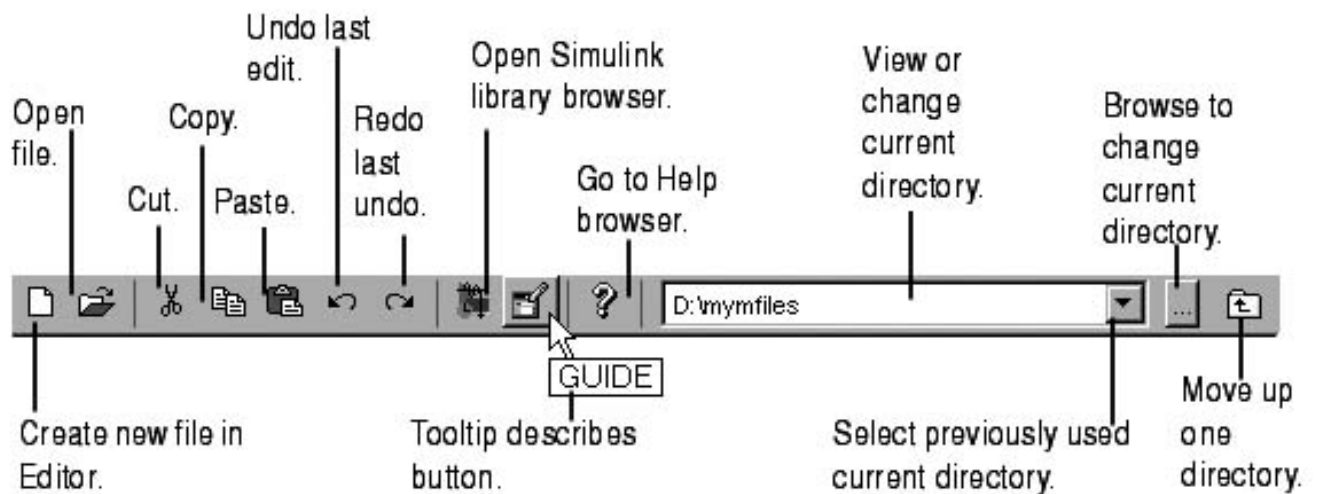


Рис.3

Назначение кнопок панели инструментов (в порядке следования слева направо):

- открыть новый файл в редакторе/отладчике
- вырезать, копировать, вставить
- отменить действие, вернуть действие
- запуск Simulink (создание новой модели Simulink)
- создание графического интерфейса пользователя (GUI)
- вызов справочной системы

### Математические выражения

Как и большинство систем программирования Matlab обеспечивает выполнение математических выражений, но в отличие от других языков программирования эти выражения содержат целые матрицы. В состав математических выражений входят:

- переменные;
- числа;
- операторы;
- функции.

### Переменная – массив/вектор/скаляр

Matlab не требует какого-либо описания типа переменной или размерности массива

Имена переменных, констант и функций могут быть составлены из любых символов алфавита, кроме специальных и цифр, начинаются с буквы. Для идентификации переменной используются первые 31 символ имени. Система различает верхний/нижний регистр букв в именах переменных и функций (как системных, так и пользовательских, SIN(x) приводит к ошибке).

Запись действительных чисел выполняется:

- в десятичной форме, знак плюс и точка у целых чисел не обязательны.
- в показательной форме по основанию десять; показатель степени отделен от мантиссы символом e или E, пробел не допускается.
- явного определения целых чисел нет;
- комплексное число представлено действительной и мнимой частями, при мнимой части проставлен символ *i* или *j* (без знака умножения);
- символ (;) после определения переменной подавляет вывод результата на экран, как и при записи оператора.

Примеры ввода переменных в систему.

```
» a=3
a =
    3
»
```

```
» a=3;
» a
a =
    3
»
```

```
» sqrt(-1)
ans =
    0 + 1.0000i
в данном случае выполнено неявное
присвоение, автоматически
использована системная переменная
ans (Answer)
```

В том случае, если переменная уже существует, ей присваивается новое значение. При задании матрицы (массива) в памяти резервируется пространство, в соответствии с ее размером.

Несмотря на выполненное выше присвоение переменной **a** численного значения, т.е. задании ее в виде скаляра, допустимо сразу или после каких-либо действий переопределить переменную, например, задать в виде матрицы:

```
» a = [1 2 3; 4 5 6; 7 8 9];
»
```

```
Или так:
» a = [1 2 3;
      4 5 6;
      7 8 9];
»
```

Диапазон представления чисел при вычислениях  $10^{-308} - 10^{308}$ , все внутренние вычисления производятся с двойной точностью.

Точность вычислений с плавающей точкой характеризует специальная системная константа  $\text{eps} = 2^{-52}$  или  $2.2204\text{e-}016$

### Операторы

Язык Matlab это язык операторов. Операторы задаются по одному в командной строке для исполнения в интерактивном режиме или в виде списка в **m**- файле или в **script**- файле. Фактически **m**- файл является программой, которая интерпретируется и выполняется системой Matlab. После выполнения **m**- файла в операционной среде системы, до завершения сеанса, остаются все значения глобальных переменных и они доступны для выполнения любых действий путем задания операторов в командной строке.

Две формы записи операторов:

- с явным присвоением: **переменная = выражение;**
- с неявным присвоением: **выражение;**

Оператор содержит:

- имена переменных и числовые константы;
- имена функций;
- специальные символы указывающие на выполняемые действия + - \* / ^ ' и на порядок действий ();
- символ (;) указывающий на завершение строки матрицы и на подавление вывода результата на экран;
- разделители операторов при записи более одного оператора в строке (,);
- символы продолжения строки для записи более 256 символов – точки, не менее двух;
- пробелы, в любых местах, они не влияют на выполняемые действия и служат для оформления строки;
- символ % указывает на то, что следующие за ним символы являются комментарием, с него можно начать строку.

Признаком начала командной строки является указатель >>, выводимый системой автоматически, при готовности к приему команды.

При записи оператора с неявным присвоением, результат вычислений присваивается автоматически внутренней переменной **ans** (answer), может быть вызван и использован по этому имени и сохраняется до выполнения следующего оператора с неявным присвоением.

Для построения выражений используются арифметические операторы и знаки, указывающие на порядок действий («help ops»).

Примеры записи переменных, скаляров и матриц, выполнение простых операторов:

```

»
» a = 3; b = -5;      % символ ; подавляет эхо-вывод
» c = (a + b)/2
c =
    -1
» pi                % встроенная константа
ans =
    3.1416
» i                 % комплексные числа, i - встроенная константа
ans =
    0 + 1.0000i
» a = 3 + 2i;       % i без знака умножения

```

### Встроенные функции

Matlab выполняет вычисление большого количества стандартных математических функций – sin, cos, exp, log, sqrt, abs и других (вызов справки с перечнем help elfun). Задание отрицательного аргумента для log, sqrt не приводит к ошибке, а автоматически вычисляется соответствующий результат в виде комплексного числа.

В системе Matlab реализованы также более сложные и специальные функции. Они сгруппированы по разделам, для их использования рекомендуется обратиться к справочной системе (help specfun, help elmat и др.)

Стандартные математические функции являются встроенными, что обеспечивает высокую скорость вычислений, однако алгоритм, описание порядка действий и вмешательство в программу недоступны.

Специальные функции реализованы в виде **m**-файлов (sinh, gamma), т.е. являются внешними. **m**-файл доступен для просмотра, анализа выполняемых операций, а также для изменений, которые необходимы для решения какой-либо конкретной задачи. **m**-файлы функций содержат в начальных строках комментарии, которые выводит справочная система при задании команды **help funname**, где **funname** – имя функции.

### Векторы и матрицы

```

» a = [1 2 3 4 5];      % ввод вектора
» sum(a)                % использование функции SUM
ans =
    15
»
» a = [1 2; 3 4];       % сложение матриц
» b = [5 6; 7 8];
» a + b
ans =
     6     8
    10    12
»

```

» c = a + b	% c присвоением переменной
c =	
6	8
10	12

```

>> a(1,1)           % задание одного элемента матрицы
ans =
    1

```

Элементы матрицы имеют индексы в соответствии с принятым в математике порядком:

$$A = \begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Bmatrix}$$

```

>> a = [1 2 3
        4 5 6
        7 8 9];
>> a(2,3)
ans =
    6

```

```

>> a
a =
    1    2
    3    4

```

```

>> a'           % транспонирование
ans =
    1    3
    2    4

```

### Функции обработки векторов

Функции	Назначение
<b>s=sum(a)</b>	Сумма всех элементов вектора <b>a</b>
<b>p=prod(a)</b>	Произведение всех элементов вектора <b>a</b>
<b>m=max(a)</b>	Нахождение максимального значения среди элементов вектора <b>a</b>
<b>[m,k]=max(a)</b>	Второй выходной аргумент <b>k</b> содержит номер максимального элемента в векторе <b>a</b>
<b>m=min(a)</b>	Нахождение минимального значения среди элементов вектора <b>a</b>
<b>[m,k]=min(a)</b>	Второй выходной аргумент <b>k</b> содержит номер минимального элемента в векторе <b>a</b>
<b>m=mean(a)</b>	Вычисление среднего арифметического элементов вектора <b>a</b>
<b>a1=sort(a)</b>	Упорядочение элементов вектора по возрастанию
<b>[a1,ind]=sort(a)</b>	Второй выходной аргумент <b>ind</b> является вектором из целых чисел от 1 до <b>length(a)</b> , который соответствует проделанным перестановкам
<b>L=length(a)</b>	Нахождение длины вектора

Указание номеров элементов вектора можно использовать и при вводе векторов, последовательно добавляя новые элементы (не обязательно в порядке возрастания их номеров). Команды:

```
>> h=10; h(2)=20; h(4)=40;
```

приводят к образованию вектора:

```

>> h
h =
10 20 0 40

```

Для ввода первого элемента **h** не обязательно указывать его индекс, т.к. при выполнении оператора **h=1** создается вектор (массив размера один на один). Следующие операторы

присваивания приводят к автоматическому увеличению длины вектора **h**, а пропущенные элементы (в нашем случае **h(3)**) получают значение ноль.

Индексация вектором служит для выделения элементов с заданными индексами в новый вектор. Индексный вектор должен содержать номера требуемых элементов, например:

```
» z=[0.2 -3.8 7.9 4.5 7.2 -8.1 3.4];
>> ind=[3 5 7];
>> znew=z(ind)
znew =
7.9000 7.2000 3.4000
```

Вектора-столбцы с одинаковым числом элементов можно складывать и вычитать друг из друга при помощи знаков "+" и "-". Такое действие верно и для векторов-строк:

```
>> c=a+b;
>> w=u-v;
```

Сложение и вычитание вектора-строки и вектора-столбца или векторов разных размеров приводит к ошибке. Операция \* предназначена для умножения векторов по правилу матричного умножения. Поскольку MatLab различает вектора-строки и вектора-столбцы, то допустимо либо умножение вектора-строки на такой же по длине вектор-столбец (скалярное произведение), либо умножение вектора-столбца на вектор-строку (внешнее произведение, в результате которого получается прямоугольная матрица). Скалярное произведение двух векторов возвращает функция **dot**, а векторное — **cross**:

```
>> s=dot(a,b)
>> c=cross(a,b)
```

Векторное произведение определено только для векторов из трех элементов.

Для операции транспонирования зарезервирован апостроф '. Если вектор содержит комплексные числа, то операция ' приводит к комплексно-сопряженному вектору. При вычислении скалярного и векторного произведений функциями **cross** и **dot** не обязательно следить за тем, чтобы оба вектора были либо столбцами, либо строками. Результат получается верный, например, при обращении **c=cross(a,b')**, только **c** становится вектором-строкой.

Для обработки матриц также существуют специальные функции, например, функции для создания стандартных матриц: **zeros**, **eye**, **ones**, **rand**, **diag** (см. **help matlab\elmat**).

### Особенности оператора умножения (деления, возведения в степень)

MatLab поддерживает два вида вычислительных операций с векторами и матрицами: матричные и поэлементные.

Наряду с умножением по правилу матричного умножения, существует операция поэлементного умножения **.\*** (точка со звездочкой). Данная операция применяется к векторам одинаковой длины и приводит к вектору той же длины, что исходные, элементы которого равны произведениям соответствующих элементов исходных векторов. Аналогично может быть выполнена операция с матрицами одинаковой размерности, в этом случае матрицы - операнды обрабатывается системой как таблицы, выполняется перемножение соответствующих элементов таблиц, результатом будет матрица такой же размерности. Например, для введенных ранее матриц **a** и **b**:

```
» a * b           % операция с матрицами
ans =
    19    22
    43    50
» a.*b           % операция с массивами (таблицами)
ans =
     5    12
    21    32
»
```

Аналогичным образом выполняется поэлементное деление **./** (точка с косой чертой). Кроме того, операция **.\** (точка с обратной косой чертой) осуществляет обратное поэлементное деление, то

есть выражения  $\mathbf{a} ./ \mathbf{b}$  и  $\mathbf{b} . \backslash \mathbf{a}$  эквивалентны. Возведение элементов вектора  $\mathbf{a}$  в степени, равные соответствующим элементам  $\mathbf{b}$ , производится с использованием  $.^{\wedge}$ . Для транспонирования векторов-строк или векторов-столбцов предназначено сочетание  $.'$  (точка с апострофом). Операции  $'$  и  $.'$  для вещественных векторов приводят к одинаковым результатам. Не обязательно применять поэлементные операции при умножении вектора на число и числа на вектор, делении вектора на число, сложении и вычитании вектора и числа. При выполнении, например, операции  $\mathbf{a} * 2$ , результат представляет собой вектор того же размера, что и  $\mathbf{a}$ , с удвоенными элементами.

### Оператор (:)

```

» a = [1 2 3 4
        5 6 7 8];
» sum(a(:,1))           % знак : => операция со столбцом
ans =
     6
» a(2,:)
ans =
     5     6     7     8
»
» 1:6                   % интервал значений целых чисел
ans =
     1     2     3     4     5     6
» a = 1:5               % то же с присвоением
a =
     1     2     3     4     5
»
» a = 1.1:5.5          % с десятичными знаками, шаг единица
a =
  1.1000  2.1000  3.1000  4.1000  5.1000
»
» a = 0.1:0.1:0.5     % дробный шаг
a =
  0.1000  0.2000  0.3000  0.4000  0.5000
»

```

Индексация двоеточием позволяет выделить идущие подряд элементы в новый вектор. Начальный и конечный номера указываются в круглых скобках через двоеточие, например:

```

» z=[0.2 -3.8 7.9 4.5 7.2 -8.1 3.4];
» znew=z(3:6)
znew =
  7.9000  4.5000  7.2000 -8.1000

```

Вызов функции **prod** с заданием интервала с помощью двоеточия вычисляет произведение элементов вектора  $\mathbf{z}$  со второго по шестой:

```

» p=prod(z(2:6))
» a = [1 2
        3 4];
» b = [5
        6];
» c = [a,b]           % добавление столбца в матрицу
c =
     1     2     5
     3     4     6

```



### Файл-функция и файл-программа

Программирование в системе MatLab осуществляется путем написания М – файлов, которые могут быть либо файл-функциями, либо файл-программами. Файл-программа является текстовым файлом с расширением m (М-файлом), в котором записаны команды и операторы MatLab.

Для запуска редактора М-файлов следует нажать кнопку New M-file на панели инструментов, либо выбрать в меню File в пункте New подпункт М-file. На экране появляется окно редактора. В нем можно набрать команды, например для построения графика:

```
x=[-1:0.01:1];
y=exp(x);
plot(x,y)
grid on
title('Экспоненциальная функция')
```

Для запуска программы следует выполнить команду Run из меню Debug в редакторе. Если предварительно не была выполнена команда Save, то система предложит сначала сохранить файл программы (Save and Run).

Для запуска программы можно набрать в командной строке имя М-файла (без расширения) и нажать <Enter>, то есть выполнить, как команду MatLab. При этом следует учесть, что путь к каталогу с М-файлом должен быть известен MatLab.

Когда текущий каталог установлен, то все М-файлы, находящиеся в нем, могут быть запущены из командной строки, либо из редактора М-файлов. Все переменные файл-программы после ее запуска доступны в рабочей среде, т. е. являются глобальными. Убедится в этом можно, выполнив команду **whos**. Файл-программа может использовать переменные рабочей среды. Например, если была введена команда:

```
>> a=[0.1 0.4 0.3 1.9 3.3];
```

то файл-программа, содержащая строку **bar(a)**, построит столбцевую диаграмму вектора **a** (разумеется, если он не был переопределен в самой файл-программе).

Файл-функции отличаются от файл-программ тем, что они могут иметь входные и выходные аргументы, а все переменные, определенные внутри файл-функции, являются локальными и не видны в рабочей среде. М-файл, содержащий файл-функцию, должен начинаться с заголовка, после него записываются операторы MatLab. Заголовок состоит из слова **function**, списка выходных аргументов, имени файл-функции и списка входных аргументов. Аргументы в списках разделяются запятой. Пример простейшей файл-функции с двумя входными и одним выходным аргументами:

```
function c=mymax(a,b)
c=a+b;
```

Практически все функции MatLab являются файл-функциями и хранятся в одноименных М-файлах. Функция **sin** допускает два варианта вызова: **sin(x)** и **y=sin(x)**, в первом случае результат записывается в **ans**, а во втором — в переменную **y**. Наша функция **mymax** ведет себя точно так же. Более того, входными аргументами **mymax** могут быть массивы одинаковых размеров или массив и число.

Разберем теперь, как создать файл-функцию с несколькими выходными аргументами. Список выходных аргументов в заголовке файл-функции заключается в квадратные скобки, сами аргументы отделяются запятой. В качестве примера ниже приведена файл-функция **quadeq**, которая по заданным коэффициентам квадратного уравнения находит его корни:

```
function [x1,x2]=quadeq(a,b,c)
D=b^2-4*a*c;
x1=(-b+sqrt(D))/(2*a);
x2=(-b-sqrt(D))/(2*a);
```

При вызове **quadeq** из командной строки используйте квадратные скобки для указания переменных, в которые будут занесены значения корней. В конце строки не ставим точку с запятой и система выдает ответ на экран в командное окно.

```
>> [r1,r2]= quadeq(1,3,2)
r1 =
-1
r2 =
-2
```

### **Справочная система (HELP)**

Для обращения к справочной системе необходимо в командном окне MATLAB набрать команду

» **help**

При этом будет представлен перечень разделов (HELP topics) справочной системы. Ниже приведены разделы, на которые следует обратить внимание в первую очередь.

- » **help matlab\ops** - выводит перечень операторов и специальных символов, используемых в системе.
- » **help arith** - об арифметических операторах,
- » **help punct** - об использовании специальных символов в командах,
- » **help colon** - о применении специального символа **:** (двоеточие), который управляет выполнением ряда важных операций с матрицами.
- » **help matlab\lang** - описание языка системы для работы в режиме интерпретации команд и программирования (написания М- файлов).
- » **help matlab\elmat**- простые матрицы и базовые операции с матрицами.
- » **help matlab\elfun** - элементарные, базовые функции системы, в том числе тригонометрические, экспоненциальные, обработки комплексных чисел и т.д.
- » **help matlab\matfun** - функции линейной алгебры и матричного анализа.
- » **help matlab\polyfun** - функции работы с полиномами и интерполяции.
- » **help matlab\plotxy** - построение графиков по двум координатным осям.

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1:** рабочая среда системы MATLAB, вычислительные операторы, задание векторов и матриц, функциональное программирование, построение графиков.

### ЗАДАНИЕ НА ЛАБОРАТОРНЫЙ ПРАКТИКУМ

1. Изучить ввод данных в системе MATLAB, выполнить примеры, приведенные выше в разделе «Методические указания».

2. Выполнить операции с векторами (исходные данные взять из таблицы 1).

Для заданных векторов **a** и **b** длины **n**:

- вычислить их сумму, разность и скалярное произведение;

- образовать вектор  $c = [a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n]$ , определить его максимальный и минимальный элементы и поменять их местами;

- упорядочить вектор **c** по возрастанию и убыванию;

- переставить элементы вектора **c** в обратном порядке и записать результат в новый вектор;

- найти векторное произведение  $u = [a_1, a_3, a_4]$  и  $v = [b_2, b_3, b_4]$ .

3. Вычислить значения функции для всех элементов матрицы и записать результат в матрицу того же размера, что и исходная матрица (варианты заданий см ниже).

4. Выполнить аппроксимацию данных полиномом и построить графики.

Задать два вектора **x** и **y** одинаковой размерности с координатами точек графика некоторой функциональной зависимости и построить график **y(x)** (функция **plot(x,y)**).

Выполнить аппроксимацию заданных точек функции по методу наименьших квадратов полиномом заданной степени ( $n = 2 - 5$ ) (функция **polyfit**).

Выполнить вычисление значений аппроксимирующего полинома в точках, определяемых аргументом - вектором **x** и занести полученные значения в вектор **z** (функция **polyval**). Построить график **z(x)** и сравнить его с заданным графиком **y(x)**.

Графики строить в одном окне (**plot(x,y, x,z)**).

Оценить качество приближения по разности  $y(x_i) - z(x_i)$ .

Оформить сделанную программу, включая построение графика, в виде подпрограммы-функции, входными параметрами которой являются вектора **x** и **y** и степень аппроксимирующего полинома. Выходной параметр – вектор коэффициентов полинома.

5. Для динамической системы (гиростабилизатор), с заданной передаточной функцией, рассчитать и построить амплитудную и фазовую частотные характеристики. Расчет выполнить с использованием комплексной переменной. Методические указания по составлению программы приведены ниже. Построить амплитудную и фазовую частотные характеристики с использованием функции **bode**. Сравнить результаты.

6. Анализ динамической системы с использованием функций Matlab.

Для гироскопического стабилизатора задать передаточную функцию, вычислить корни характеристического полинома (**roots**).

Командой **ltiview** вызвать интерактивный обозреватель свойств линейных моделей (**LTViewer**) и выполнить анализ свойств гиростабилизатора.

Параметры гиростабилизатора взять из курсовой работы.

**Таблица 1** (варианты данных к п. 2 задания)

$a = [0.5 \ 3.7 \ 6.0 \ -4.3 \ 1.2 \ -2.7 \ 2.4 \ 2.2];$	$b = [3.6 \ 7.0 \ 7.0 \ 5.4 \ 2.6 \ -2.7 \ -6.4 \ 0.3].$
$a = [-4.8 \ -1.3 \ -1.0 \ 0.7 \ 4.0 \ 5.8 \ 4.3 \ -8.0];$	$b = [-1.1 \ -1.9 \ 7.1 \ -2.1 \ 6.8 \ 2.8 \ 0.3 \ 1.6].$
$a = [1.0 \ -3.9 \ -2.3 \ -3.3 \ -1.7 \ 2.2 \ -0.6 \ 1.8];$	$b = [2.7 \ -2.7 \ -2.2 \ 4.4 \ 0.4 \ -6.0 \ -3.4 \ -5.2].$
$a = [-2.4 \ 3.3 \ -0.1 \ 3.6 \ 7.4 \ -2.8 \ 0.3 \ 2.2];$	$b = [6.3 \ 0.6 \ 4.3 \ -3.7 \ -7.0 \ 3.7 \ 3.7 \ 8.0].$
$a = [8.4 \ -5.9 \ -6.5 \ -0.9 \ 6.9 \ -1.7 \ 1.7 \ 0.8];$	$b = [-0.0 \ 2.0 \ -1.5 \ 7.5 \ -4.0 \ -3.0 \ -6.2 \ 0.0].$
$a = [5.3 \ 6.8 \ -7.1 \ 6.8 \ -4.0 \ -2.3 \ -4.4 \ -0.2];$	$b = [7.5 \ -1.5 \ -4.9 \ -4.6 \ -2.3 \ -5.3 \ 5.5 \ 2.3].$
$a = [1.2 \ -4.1 \ -0.8 \ -0.7 \ -2.2 \ 1.7 \ 3.3 \ -6.1];$	$b = [-1.5 \ 2.2 \ 1.0 \ -4.3 \ -0.0 \ -1.8 \ -1.5 \ 2.4].$
$a = [6.6 \ -5.0 \ -2.7 \ 8.3 \ 3.8 \ 1.9 \ 1.1 \ 2.7];$	$b = [-1.0 \ 3.2 \ 4.2 \ -6.4 \ 1.9 \ -6.5 \ -6.2 \ -8.1].$
$a = [-1.9 \ 0.4 \ 1.8 \ 4.2 \ -3.8 \ -4.7 \ 4.0 \ -2.1];$	$b = [-8.7 \ -4.2 \ -1.4 \ 2.8 \ -2.2 \ 7.8 \ 0.0 \ -0.1].$
$a = [0.9 \ 1.7 \ -3.2 \ -3.8 \ 7.3 \ 6.0 \ -0.2 \ 8.6];$	$b = [0.6 \ -0.4 \ -6.9 \ -2.2 \ 1.6 \ 3.8 \ -3.2 \ 0.4].$

**Таблица 2** (варианты данных к п. 3 задания)

1. $f(x) = x^3 - 2x^2 + \sin x - 4;$	$A = \begin{bmatrix} 9.33 & -4.01 & 8.19 & 2.64 \\ 0.55 & 3.81 & 3.32 & 5.07 \end{bmatrix}.$
2. $f(x) = \frac{e^x - x}{e^x + x};$	$A = \begin{bmatrix} 9.32 & 0.21 & -9.89 & 3.11 \\ 0.54 & 4.99 & 5.01 & -0.03 \end{bmatrix}.$
3. $f(x) = \sqrt{1 + \sqrt{ x ^3 + 1}};$	$A = \begin{bmatrix} -1.54 & 0.49 & 3.11 & 2.99 \\ 4.05 & -5.85 & 3.72 & 0.11 \end{bmatrix}.$
4. $f(x) = e^x \sin x - e^{-x} \cos x;$	$A = \begin{bmatrix} -9.04 & 3.36 & 3.09 & -2.49 \\ -4.33 & -5.09 & 9.74 & 1.65 \end{bmatrix}.$
5. $f(x) = \ln( x ) \sin \pi x;$	$A = \begin{bmatrix} 0.33 & 0.95 & 7.12 & -9.22 \\ -0.64 & 3.76 & 1.34 & -0.03 \end{bmatrix}.$
6. $f(x) = e^{x^2+x+1};$	$A = \begin{bmatrix} -4.53 & -2.12 & -6.54 & -3.21 \\ 3.43 & 7.43 & -0.25 & 1.64 \end{bmatrix}.$
7. $f(x) = \frac{\sqrt[3]{x^2 - 1}}{ x  + 3};$	$A = \begin{bmatrix} 0.23 & 3.89 & -4.23 & -7.25 \\ 5.84 & 5.13 & -0.89 & 3.55 \end{bmatrix}.$
8. $f(x) = \frac{1}{1 + \frac{1+x}{1-x^2}};$	$A = \begin{bmatrix} -5.84 & 9.84 & 0.23 & 1.59 \\ -9.25 & -0.25 & 1.54 & 0.43 \end{bmatrix}.$

$$9. \quad f(x) = \frac{x^3 + \sin x}{x^3 - \cos x} \sqrt{e^x + 1}; \quad A = \begin{bmatrix} 0.64 & 6.34 & 0.32 & -4.23 \\ 1.19 & 3.23 & 1.54 & 0.43 \end{bmatrix}.$$

$$10. \quad f(x) = \arcsin(\cos x^2); \quad A = \begin{bmatrix} \pi & 2.2\pi & -2\pi & 0.3\pi \\ 3\pi & -\pi & 0.1\pi & 5\pi \end{bmatrix}.$$

**Методические указания к п. 5.** (пример расчета амплитудной и фазовой частотной характеристики)

Исходные данные заданы в виде коэффициентов полиномов числителя и знаменателя передаточной функции в порядке убывания степени оператора  $p$ .

Данный пример основан на непосредственной подстановке в формулу передаточной функции комплексной частоты ( $j\omega$ ) и показывает принципы использования вычислений с комплексными числами и функциями системы Matlab. Для выполнения полученного задания необходимо модифицировать программу в соответствии с видом передаточной функции.

```
% Frequency response
%      1      a(p)
%  W(p) = ----- = -----
%      Tp + 1  b(p)
% Numerator
%      a=[ 1 ];
% Denominator
%      b= [ 3 1 ];
w=logspace(-2,3,100);
s=sqrt(-1)*w(:);
g=polyval(b,s).\polyval(a,s);
mag=abs(g)';
phase = (180./pi)*angle(g)';
figure(1)
semilogx(w,20*log(mag));
figure(2)
semilogx(w,phase)
```

Используемые функции системы MATLAB:

**logspace** - формирование вектора с логарифмическим распределением набора точек аргумента в заданном диапазоне частот.

**sqrt**- вычисление квадратного корня.

**polyval** - вычисление значений полинома в заданных точках.

**abs**- модуль.

**angle** – угол в радианах.

**semilogx** - график с логарифмическим масштабом по оси  $x$ .

**log** - логарифм десятичный.

По завершении работы программы все переменные сохраняются в оперативной памяти компьютера с последними присвоенными им значениями. Это позволяет просмотреть их или использовать для вычислений по командам, непосредственно вводимым в командной строке. Например, можно вычислить корни полиномов числителя или знаменателя заданной передаточной функции: **roots(b)**

На рисунок графика нанести сетку командой **grid**.

## Методические указания к п. 6.

Зададим LTI модель передаточной функцией, например

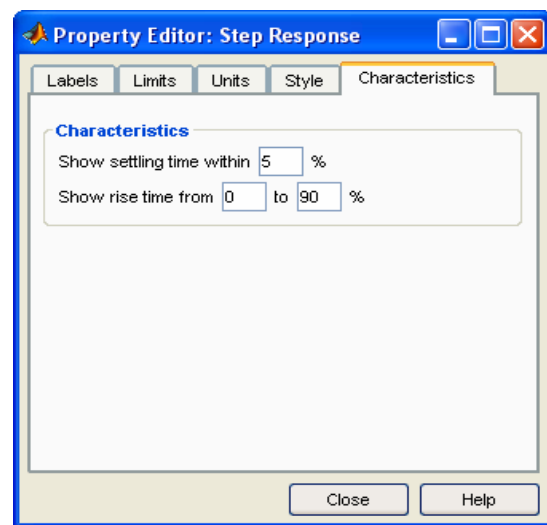
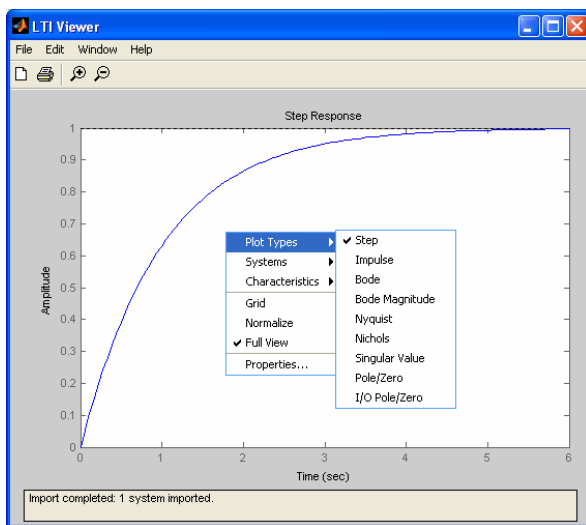
```
>> w=tf([1],[1 1])
Transfer function:
1
```

```
-----
s + 1
```

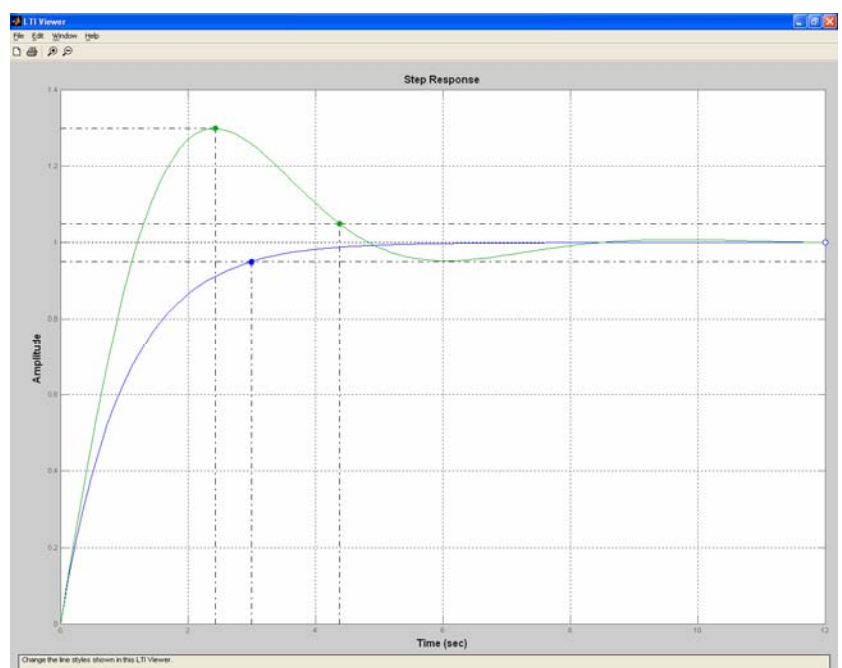
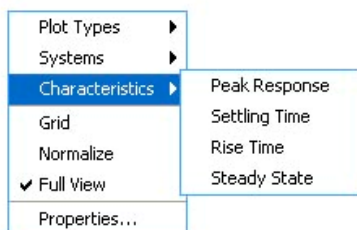
затем вызываем LTIviewer командой

```
>> ltiview
```

При этом появляется рабочее окно с заготовкой графика. По команде меню FILE – IMPORT появляется окно, представляющее содержимое рабочей области, в данном случае там присутствуют для объекта – системы w и w1. Выбрав объект, нажимаем ОК и смотрим результат на графике. Правая кнопка мыши открывает выпадающее меню, позволяющее выполнить настройку графиков, в том числе выбрать требуемую характеристику системы (реакция на ступенчатый или импульсный входной сигнал, ЛАФЧХ и др.) и другие параметры обозревателя.



В обозревателе можно загрузить несколько систем, в данном случае это w и w1. заданные характеристики переходного процесса рассчитываются автоматически. Параметры задаются в выпадающем меню **Properties**.



Результат представлен на графике.