Казанский национальный исследовательский
технический университет им. А. Н. Туполева-КАИ

Институт радиоэлектроники и телекоммуникаций

Кафедра радиоэлектронных и телекоммуникационных систем

Методические указания к лабораторной работе «Основы работы с MATLAB»

по дисциплине
«Прикладные информационные технологии»

Казань, 2015 г.

Лабораторная работа № 1

Цель работы: Изучение основ работы с системой MATLAB, получение навыков решения базовых математических задач в среде MATLAB.

Введение

MATLAB – система многоцелевого назначения, которая вышла на рынок программных продуктов почти двадцать лет назад и с тех пор непрерывно совершенствовалась. Но первоначально ее основу составляли алгоритмы решения систем линейных уравнений и задач на собственные значения, откуда и произошло ее название «матричная лаборатория». Теперь она представляется наиболее эффективной при проведении прикидочных расчетов и при разработке новых алгоритмов. Сейчас уже существует несколько десятков специальных приложений к MATLAB, посвященных более узким проблемам. Это обработка сигналов и изображений, инженерное программирование в виде блок-схем, решение экономических задач и многое другое. Но любое из этих приложений можно изучать только после первоначального освоения MATLAB.

1. Основы работы с MATLAB

Среда MATLAB включает интерпретатор команд на языке высокого уровня, графическую систему, пакеты расширений и реализована на языке C. Вся работа организуется через командное окно (Command Window), которое появляется при запуске программы matlab.exe. В процессе работы данные располагаются в памяти (Workspace), для изображения кривых, поверхностей и других графиков создаются графические окна.

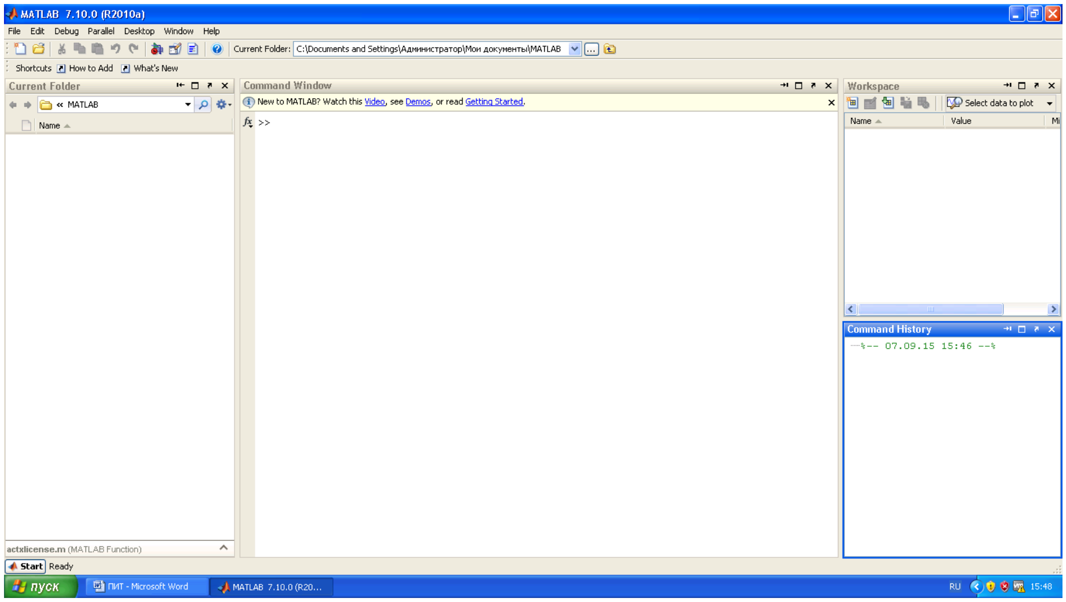


Рис 1.1 Внешний вод командного окна MATLAB

В командном окне в режиме диалога проводятся вычисления. Пользователь вводит команды или запускает на выполнение файлы с текстами на языке MATLAB. Интерпретатор обрабатывает введенное и выдает результаты: числовые и строковые данные, предупреждения и сообщения об ошибках. Строка ввода помечена знаком >>. В командном окне показываются вводимые с клавиатуры числа, переменные, а также результаты вычислений. Имена переменных должны начинаться с буквы. Знак = соответствует операции присваивания. Нажатие клавиши *Enter* заставляет систему вычислить выражение и показать результат.

Пример:

```
>> a=10+20+5
```

После нажатия клавиши *Enter* получим результат вычисления

```
a =
   35
>>
```

Все значения переменных, вычисленные в течение текущего сеанса работы, сохраняются в специально зарезервированной области памяти компьютера, называемой рабочим пространством системы MATLAB (Workspace). Командой *clc* можно стереть содержимое командного окна, однако это не затронет содержимого рабочего пространства. Когда исчезает необходимость в хранении ряда переменных в текущем сеансе работы, их можно стереть из памяти компьютера командой *clear* или *clear*(*имя1*, *имя2*, …). Первая команда удаляет из памяти все переменные, а вторая – переменные с именами *имя1* и *имя2*. Командой *who* можно вывести список всех переменных, входящих в данный момент в рабочее пространство системы. Для просмотра значения любой переменной из текущего рабочего пространства системы достаточно набрать ее имя и нажать клавишу *Enter*.

После окончания сеанса работы с системой MATLAB все ранее вычисленные переменные теряются. Чтобы сохранить в файле на диске компьютера содержимое рабочего пространства системы MATLAB, нужно выполнить команду меню File → Save Workspace As … .По умолчанию расширение имени файла mat, поэтому такие файлы принято называть MAT-файлами. Для загрузки в память компьютера ранее сохраненного на диске рабочего пространства нужно выполнить команду меню: File → Load Workspace… .

2. Элементарные математические функции MATLAB

Система MATLAB имеет в своём составе широчайший набор встроенных функций. Наиболее часто используемые элементарные функции приведены в таблицах ниже.

Таблица 2.1 Тригонометрические и гиперболические функции MATLAB

Функция	*Описание*
sin(x)	Синус, аргумент в радианах
sind(x)	Синус, аргумент в градусах
cos (x)	Косинус, аргумент в радианах
cosd (x)	Косинус, аргумент в градусах
tan(x)	Тангенс, аргумент в радианах
tand(x)	Тангенс, аргумент в градусах

Функция	*Описание*
cot(x)	Котангенс, аргумент в радианах
cotd(x)	Котангенс, аргумент в градусах
sec(x)	Секанс, аргумент в радианах
secd(x)	Секанс, аргумент в градусах
csc(x)	Косеканс, аргумент в радианах
cscd(x)	Косеканс, аргумент в градусах
asin(x)	Арксинус (для действительных чисел из диапазона [–1,1] возвращает действительные числа в диапазоне от $-\pi/2$ до $\pi/2$, а для действительных чисел вне этого диапазона - комплексные значения)
acos(x)	Арккосинус (для действительных чисел из диапазона [–1,1] возвращает действительные числа в диапазоне от 0 до π), а для действительных чисел вне этого диапазона - комплексные значения)
atan(x)	Арктангенс (для действительных чисел возвращает значения в диапазоне от $-\pi/2$ до $\pi/2$)
acot(x)	Аркотангенс
asec(x)	Арксеканс
acscd(x)	Арккосеканс
sinh(x)	Гиперболический синус
cosh(x)	Гиперболический косинус
tanh(x)	Гиперболический тангенс
coth(x)	Гиперболический котангенс
sech(x)	Гиперболический секанс
csch(x)	Гиперболический косеканс
asinh(x)	Гиперболический арксинус
acosh(x)	Гиперболический арккосинус
atanh(x)	Гиперболический арктангенс
acoth(x)	Гиперболический арккотангенс
asech(x)	Гиперболический арксеканс
acsch(x)	Гиперболический арккосеканс

Таблица 2.2 Экспоненциальные функции MATLAB

Функция	*Описание*
exp(x)	Экспонента числа x
log(x)	Натуральный логарифм
log10(x)	Десятичный логарифм
log2(x)	Логарифм по основанию 2
pow2(x)	Возведение числа 2 в степень x

Функция	*Описание*
a^x	Возведение числа а степень x
sqrt(x)	Квадратный корень
nextpow2(x)	Степень, в которую нужно возвести число 2, чтобы получить ближайшее число, большее или равное x; если x — вектор, эта функция возвращает значение, равное значению функции nextpow2(length(x))
expm1(x)	Вычисление точного значения выражения exp(x)-1 для маленьких значений x
log1p(x)	Вычисление точного значения выражения log(1+x) для маленьких значений x
nthroot(x, n)	Вычисление вещественного корня степени n для вещественного значения x (значения x и n должны быть вещественными, n — скаляром)

Таблица 2.3 Функции комплексного аргумента MATLAB

abs(x)	Возвращает модуль комплексного числа x
angle(x)	Возвращает аргумент комплексного числа x (в радианах, в диапазоне от $-\pi$ до π)
conj(x)	Возвращает число, комплексно-сопряженное относительно x
imag(x)	Возвращает мнимую часть комплексного числа x
real(x)	Возвращает действительную часть комплексного числа x
complex(A,B)	Создает комплексное число A+Bi по заданной действительной и мнимой части (A и B могут быть скалярами либо массивами одинакового размера и типа)

Таблица 2.4 Функции округления и вычисления остатка от деления

fix(x)	Округление до ближайшего целого в сторону нуля
floor(x)	Округление до ближайшего целого в сторону отрицательной бесконечности
ceil(x)	Округление до ближайшего целого в сторону положительной бесконечности
round(x)	Округление до ближайшего целого
mod(x)	Остаток от целочисленного деления с учетом знака
rem(x)	Остаток от целочисленного деления по модулю
sign(x)	Определение знака числа

3. Формирование векторов и матриц

MATLAB специально предназначен для проведения сложных вычислений с векторами и матрицами. При этом по умолчанию предполагается, что каждая переменная – это вектор или матрица. Например, если задано $x = 1$, то это значит, что x – это вектор с одним элементом, равным 1. Если надо задать вектор из трех элементов, то их значения надо перечислить в квадратных скобках, разделяя пробелами.

```
>>V = [1 2 3]
V =
        1 2 3
```

В данном случае задан вектор-строка. Если разделить элементы точкой с запятой, то получим вектор-столбец.

```
>>V = [1; 2; 3]
V =
        1
        2
        3
```

Задание матрицы требует указания несколько строк. Для разграничения строк используется символ **;** (точка с запятой).

```
>>T = [1 2 3; 4 5 6; 7 8 9]
T =
        1 2 3
        4 5 6
        7 8 9
```

Для указания отдельного элемента вектора или матрицы используются выражения вида V(i) или T(i, j). Например:

```
>>T (3,2)
ans =
        8
```

Если элементу T(i, j) нужно присвоить новое значение x, то используют оператор присваивания

```
T (3,2) = x;
```

Выражение T(i) с одним индексом дает доступ к элементам матрицы, развернутым *в один столбец*. Такая матрица образуется из исходной, если подряд выписать ее столбцы. Например:

```
>>T (3)
ans =
7
>>T (8)
ans =
        6
```

Для того чтобы задать ряд значений используется следующее выражение

```
>> x=0:2:20;
>> x
x =
   0    2    4    6    8   10   12   14   16   18   20
```

Таким образом задаётся ряд значений от нуля до двадцати с шагом 2: (удобно использовать при построении графиков и.т.д.)

Наряду с операциями над отдельными элементами матриц и векторов MATLAB позволяет производить арифметические операции сразу над всеми элементами. Для этого перед знаком операции *ставится точка*. Имеются также ряд особых функций для задания векторов и матриц. Отметим функции *ones* и *zeros*. Эти функции служат для создания одномерных и многомерных массивов. Функция *ones* создает массив с единичными элементами

```
>> a = ones (3, 2)
a =
        1  1
        1  1
        1  1
```

Функция *zeros* создает массив с нулевыми элементами

```
>> b = zeros (2, 3)
b =
0 0 0
0 0 0
```

В краевых задачах поиска частот собственных колебаний и критических сил потери устойчивости упругих систем функция *zeros* находит широкое применение, что существенно упрощает тексты соответствующих программ.

4. Операции над матрицами и векторами

В системе MATLAB достаточно просто выполняются математические операции над матрицами и векторами. Рассмотрим сначала простые операции сложения и умножения матриц и векторов. Пусть даны два вектора, Вектор строка и вектор столбец. В соответствии с операциями над векторами, умножение вектор-строки на вектор-столбец дает число, а умножение вектор-столбца на вектор-строку дает двумерную матрицу.

```
>> a = [1 2 3 4 5];
>> b = [1; 1; 1; 1; 1];
>> a*b

ans =

    15

>> b*a

ans =
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
     1     2     3     4     5
```

Сложение и вычитание двух векторов записывается так

```
>> a1=[1,2,3,4,5];

>> a2=[5,4,3,2,1];

>> a1+a2

ans =

     6     6     6     6     6
```

```
>> a1-a2

ans =

    -4    -2     0     2     4
```

Следует обратить внимание, что операции сложения и вычитания можно выполнять между двумя векторами-столбцами или двумя векторами-строками. Иначе MATLAB выдаст сообщение об ошибке, т.к. разнотипные векторы складывать нельзя. Так обстоит дело со всеми недопустимыми арифметическими операциями: в случае невозможности их вычисления система MATLAB сообщит об ошибке и выполнение программы будет завершено на соответствующей строке.

Скалярное произведение двух векторов можно найти с помощью функции *dot(a,b)*;

```
>> a=[1 2 4];

>> b=[1 3 5];

>> c=dot(a,b);

>> c

c =

   27
```

Векторное произведение векторов можно найти с помощью функции *cross(a, b)*

```
>> a=[1 2 4];

>> b=[1 3 5];

>>c= cross(a, b);

>> c

c =

   -2   -1    1
```

Аналогичным образом выполняются операции умножения и сложения между матрицами:

```
>>A = [1 2 3; 4 5 6; 7 8 9];
>>B = ones(3);
>>C = A+B;            % сложение двух матриц одинакового размера
>>D = A+5;            % сложение матрицы и числа
>>E = A*B;            % умножение матрицы А на В
>>F = B*A;            % умножение матрицы В на А
>>G = 5*A;            % умножение матрицы на число
```

Операции вычисления обратной матрицы, а также транспонирования матриц и векторов, записываются следующим образом:

```
>>a = [1 1 1];                  % вектор-строка
>>b = a';                       % вектор-столбец, образованный
                                % транспонированием вектора-строки а.


>>A = [1 2 3; 4 5 6; 7 8 9];    % матрица 3х3 элемента
>>B = a*A;                      % B = [12 15 18] – вектор-строка
>>C = A*b;                      % C = [6; 15; 24] – вектор-столбец
>>D = a*A*a';                   % E – транспонированная матрица А
>>F = inv(A);                   % F – обратная матрица А
>>G = A^-1;                     % G – обратная матрица А
```

Из приведенного примера видно, что операция транспонирования матриц и векторов обозначается символом ‘ (апостроф), который ставится после имени вектора или матрицы. Вычисление обратной матрицы можно делать путем вызова функции inv() или возводя матрицу в степень -1. Результат в обоих случаях будет одинаковым, а два способа вычисления сделано для удобства использования при реализации различных алгоритмов.

Если в процессе вычислений требуется поэлементно умножить, разделить или возвести в степень элементы вектора или матрицы, то для этого используются операторы:

.*- поэлементное умножение;

./ и - поэлементные деления;

.^ - поэлементное возведение в степень.

Рассмотрим работу данных операторов на следующем примере.

```
>>a = [1 2 3];         % вектор-строка
>>b = [3 2 1];         % вектор-строка
>>a.*b

 ans =

     3     4     3


A = ones(3);                          % матрица 3x3, состоящая из единиц
B = [1 2 3;4 5 6; 7 8 9];              % матрица 3x3

>> A.*B

 ans =

     1     2     3

     4     5     6

     7     8     9

>>A./B

ans =

    1.0000    0.5000    0.3333

    0.2500    0.2000    0.1667

    0.1429    0.1250    0.1111

>> A.*2

ans =

     2     2     2

     2     2     2

     2     2     2
```

Определитель матрицы можно найти с помощью функции *det(M)*

```
>> B=[1 2; 3 5];

>> det(B)
```

```
ans =

   -1.0000
```

Для поиска максимального значения элемента вектора используется стандартная функция max(), которая возвращает найденное максимальное значение элемента и его позицию (индекс):

```
>>a = [1 6 3 4];
>>[v, i] = max(a);

V=

      3

i=

      1

или

>>max(a);

ans =

      3
```

Приведенный пример показывает два разных способа вызова функции max(). В первом случае определяется и максимальное значение элемента и его индекс в векторе, а во втором – только максимальное значение элемента.

В случае с матрицами, данная функция определяет максимальные значения, стоящие в столбцах, как показано ниже в примере:

```
>> A=[1 5 10; 100 2 5; 3 200 9]

A =

     1     5    10

   100     2     5

     3   200     9

>> max(A)

ans =

   100   200    10
```

```
>> [V ,I]=max(A)

V =

   100   200    10

I =

     2     3     1
```

Полный синтаксис функции *max()* можно узнать, набрав в командном окне MATLAB команду.

```
help <название функции>
```

Аналогичным образом работает функция *min()*, которая определяет минимальное значение элемента вектора или матрицы и его индекс.

Другой полезной функцией работы с матрицами и векторами является функция *sum()*, которая вычисляет сумму значений элементов вектора или столбцов матрицы:

```
>> a = [3 5 4 2 1];

>> sum(a)

ans =

    15

>> A = [4 3 5; 6 7 2; 3 1 8];

>> S1 = sum(A);

>> S2 = sum(sum(A));

S1 =

    13    11    15

S2 =

      39
```

При вычислении суммы S2 сначала вычисляется сумма значений элементов матрицы A по столбцам, а затем, по строкам. В результате, переменная S2 содержит сумму значений всех элементов матрицы A.

Для сортировки значений элементов вектора или матрицы по возрастанию или убыванию используется функция sort() следующим образом:

```
a = [3 5 4 2 1];

>> a = [3 5 4 2 1];

>> b1=sort(a);

>> b2=(a, 'descend');

>> b3 = sort(a, 'ascend');

b1 =

     1     2     3     4     5

>> b2

b2 =

     5     4     3     2     1

b3 =

     1     2     3     4     5
```

для матриц

```
>> A = [4 3 5; 6 7 2; 3 1 8];

>> B1 = sort(A);

>> B2 = sort(A, 'descend');

B1 =

     3     1     2

     4     3     5

     6     7     8
```

```
B2 =

     6     7     8
     4     3     5
     3     1     2
```

Во многих практических задачах часто требуется найти определенный элемент в векторе или матрице. Это можно выполнить с помощью стандартной функции *find()*, которая в качестве аргумента принимает условие, в соответствии с которым и находятся требуемые элементы, например

```
>> a = [3 5 4 2 1];
b1 = find(a == 2);          % b1 = 4 - индекс элемента 2
b2 = find(a ~= 2);          % b2 = [1 2 3 5] - индексы без 2
b3 = find(a > 3);           % b3 = [2 3]
b1 =
     4
b2 =
     1     2     3     5
b3 =
     2     3
```

В приведенном примере символ '= =' означает проверку на равенство, а символ '~=' выполняет проверку на неравенство значений элементов вектора а.

5. Построение графиков функции в MARLAB

Графические возможности системы MATLAB являются мощными и разнообразными. Изучим наиболее простые в использовании возможности (высокоуровневую графику).

Сформируем два вектора x и y:

Для построения линейного графика необходимо вызвать функцию:

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

MATLAB показывает графические объекты в специальных графических окнах, имеющих в заголовке слово Figure.

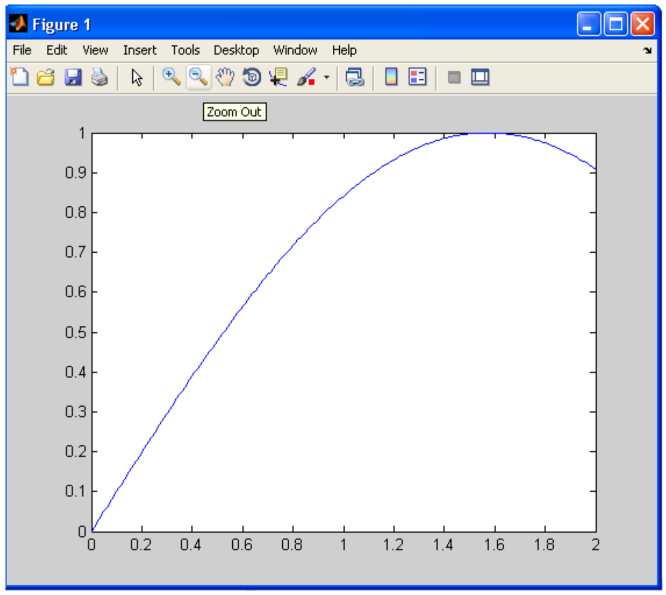


Рис 5.1

Если нужно второй график провести «поверх первого графика», то перед вторичным вызовом графической функции plot нужно выполнить команду hold on, которая предназначена для удержания текущего графического окна:

```
>> z=cos(x);

>> hold on

>> plot(x,z)
```

Практически тоже самое получится (рис. 5.2), если набрать:

```
» x=0:0.01:2; y=sin(x); z=cos(x);
» plot(x,y,x,z)
```

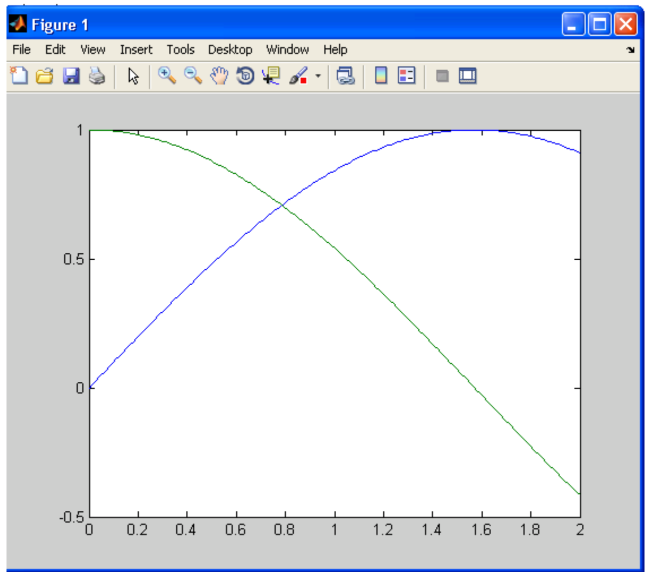


Рис 5.2 Графики функций y=sin(x), z=cos(x), построенные в одном графическом окне

Если нужно одновременно визуализировать несколько графиков так, чтобы они не мешали друг другу, то это можно сделать двумя способами. Первым решением является построение их в разных графических окнах. Для этого перед вторичным вызовом функции plot следует набрать команду figure, которая создает новое графическое окно и заставляет все последующие за ней функции построения графиков выводить их туда. Вторым решением показа нескольких графиков без конфликта диапазонов осей координат является использование функции subplot. Эта функция позволяет разбить область вывода графической информации на несколько подобластей, в каждую из которых можно вывести графики различных функций.

Например, для ранее выполненных вычислений с функциями sin и cos постройте графики этих двух функций в первой подобласти, а график функции exp(x) – во второй подобласти одного и того же графического окна (рис. 5.3):

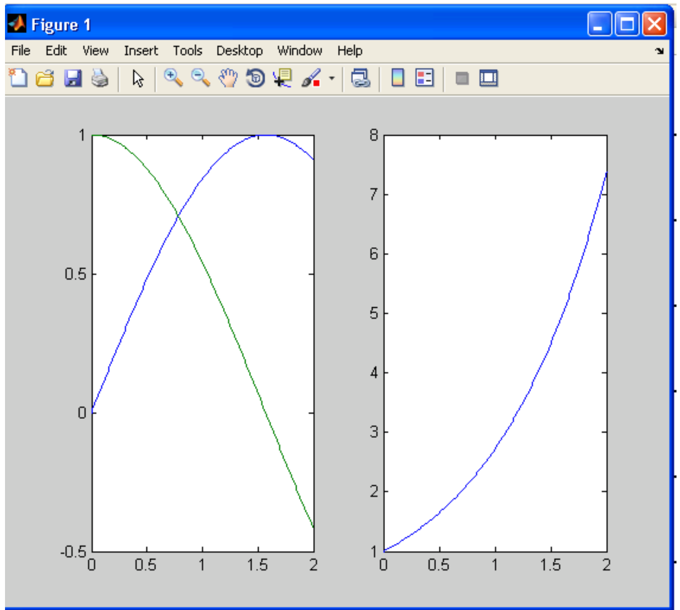


Рис. 5.3 Графики функций y=sin(x), z=cos(x) и w=exp(x), построенные в двух подобластях одного графического окна

Диапазоны изменения переменных на осях координат этих подобластей независимы друг от друга. Функция *subplot* принимает три числовых аргумента, первый из которых равен числу рядов подобластей, второй равен числу колонок подобластей, а третий аргумент – номеру подобласти (номер отсчитывается вдоль рядов с переходом на новый ряд по исчерпании). Снять действие функции *subplot* можно командой:

```
>>subplot(1,1,1)
```

Построить график функции в полярных координатах (рис. 5.4) можно с помощью графической функции polar.

```
>> phi=0:0.01:2*pi; r=sin(3*phi);
>> polar(phi,r)
```

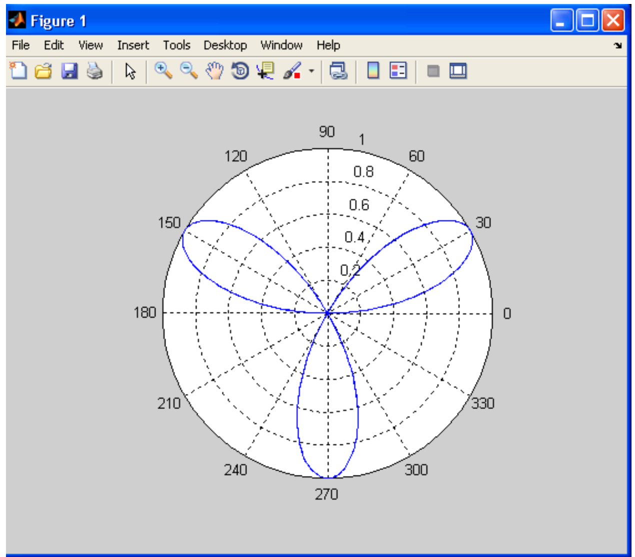


Рис. 5.4. График функции r=sin(3*phi) в полярных координатах

Для проставления различных надписей на полученном рисунке применяют функции xlabel, ylabel, title и text. Функция xlabel создает подпись у горизонтальной оси, функция ylabel – тоже для вертикальной оси (причем эти надписи ориентированы вдоль осей координат). Если требуется разместить надпись в произвольном месте рисунка, применяем функцию text. Общий заголовок для графика создается функцией title. Кроме того, используя команду grid on, можно нанести измерительную сетку на всю область построения графика. Например (Рис. 5.5).

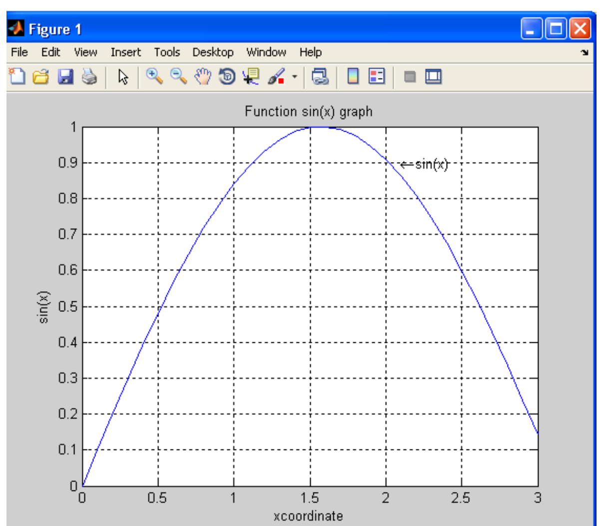


Рис. 5.5. График функции y=sin(x), построенный в с надписями на координатных осях и на рисунке

```
» x=0:0.1:3; y=sin(x);
» plot(x,y);
» title('Function sin(x) graph');
» xlabel('xcoordinate'); ylabel('sin(x)');
» text(2.1, 0.9, '\leftarrowsin(x)'); grid on
```

Надпись функцией text помещается начиная от точки с координатами, указанными первыми двумя аргументами. По умолчанию координаты задаются в тех же единицах измерения, что и координаты, указанные на горизонтальной и вертикальной осях. Специальные управляющие символы вводятся внутри текста после символа \ (обратная косая черта).

Возможности отображения трехмерных графических объектов в системе MATLAB весьма обширны. Мы сосредоточимся на изображении пространственных линий и на построении графиков функций двух вещественных переменных, которые представляют поверхности в пространстве. Каждая точка в пространстве характеризуется тремя координатами. Набор точек, принадлежащих некоторой линии в пространстве, нужно задать в виде трех векторов, первый из которых содержит первые координаты этих точек, второй вектор – вторые их координаты, ну а третий вектор - третьи координаты. После чего эти три вектора можно подать на вход функции *plot3*, которая и осуществит проектирование соответствующей трехмерной линии на плоскость и построит результирующее изображение (рис. 5.6). Например:

```
>> t=0:pi/50:10*pi;
>> x=sin(t);
>>y=cos(t); plot3(x,y,t); grid on
```

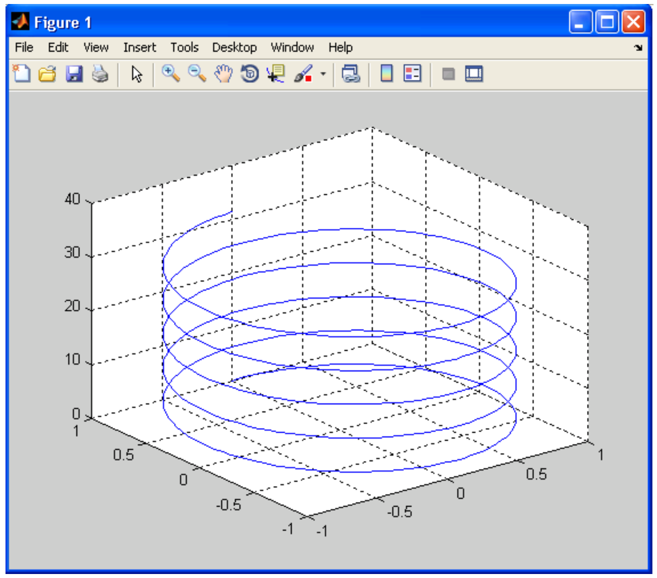
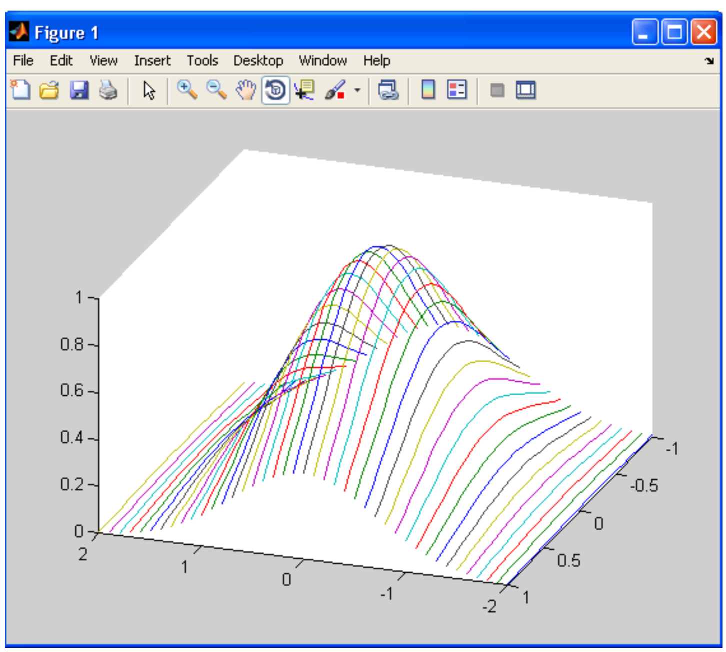


Рис. 5.6. График винтовой линии, построенный с помощью функции *plot3*

Эту же функцию *plot3* можно применить и для изображения поверхностей в пространстве, если, конечно, провести не одну линию, а много.

```
>> u=-2:0.1:2; v=-1:0.1:1;
>> [X,Y]=meshgrid(u,v);
>> z=exp(-X.^2-Y.^2);
>> plot3(X,Y,z)
```



6. Поиск всех корней полинома

При решении прикладных задач очень часто возникает необходимость найти все корни полинома (включая комплексные). В MATLAB для этой задачи предусмотрена специальная функция *roots()*.

Пример. Требуется найти корни полинома

$$P(x) = 3x^6 + 2x^4 - x^2 + 4x + 1$$

Для этого создаем вектор p его коэффициентов (указывая все коэффициенты заданного полинома, включая нулевые):

```
>> p = [3 0 2 0 -1 4 1];
```

и вызываем функцию roots, которая возвращает вектор корней полинома

```
>> r = roots(p)
```

Получаем, что у данного полинома шестой степени два вещественных корня и две пары комплексно-сопряженных, т.е. всего шесть корней.

```
r =
   0.79356 +    0.62699i
   0.79356 -    0.62699i
  -0.20431 +     1.1899i
  -0.20431 -     1.1899i
  -0.94089
  -0.23761
```

Требования к отчёту

1.Выполнить задания согласно варианта.

2.Все команды результаты, и графики скопировать из командного окна в текстовый файл MS Word.

3.Представить отчёт преподавателю в печатном или электронном виде.

Отчёт можно отправить на электронный адрес lesha_ulanov@mail.ru с указанием фамилии и группы.

Задание 1

Найти значение функции $f(x)$, при $x = x_0$, Построить график функции $f(x)$, оси подписать

Варианты заданий.

№ варианта	Функция	x_0
1	$y = \dfrac{e^{-x}}{\sqrt{e^{-x}+1}} - x$	5,5
2	$y = \dfrac{x^2-1}{\ln(x^2-1)} + x$	2,75
3	$y = shx + \sin x - 1$	3,1
4	$y = \dfrac{1}{x(1-\ln x)} - 2$	4,21
5	$y = \dfrac{2}{3}\sin^2 2x - \dfrac{3}{4}\cos^2 2x$	6,32
6	$y = \dfrac{\sin^2 x}{\cos^4 x} - 1$	4,75
7	$y = e^x \sqrt[3]{x^2} - x - 1$	2,35
8	$y = x^3 \sqrt[3]{(1-x)^2} - 1$	8,29
9	$y = e^{-x}\sqrt{1+x+x^2} - x^2$	4,56
10	$y = \sqrt{x} - 1 - \cos(0{,}5x)$	1,23

Задание 2

Найти скалярное и векторное произведение векторов $\overline{a}$ и $\overline{b}$

Вариант	$\overline{a}$ ={1, 2, 4}	$\overline{b}$ ={21, 44, 3}
1	$\overline{a}$ ={3, 8, 5}	$\overline{b}$ ={22, 21, 1}
2	$\overline{a}$ ={1, 5, 3}	$\overline{b}$ ={2, 3, 5}
3	$\overline{a}$ ={4, 5, 1}	$\overline{b}$ ={8, 9, 13}
4	$\overline{a}$ ={9, 11, 13}	$\overline{b}$ ={32, 21, 17}
5	$\overline{a}$ ={4, 9, 17}	$\overline{b}$ ={19, 15, 13}
6	$\overline{a}$ ={3, 15, 13}	$\overline{b}$ ={33, 56, 18}
7	$\overline{a}$ ={11, 32, 5}	$\overline{b}$ ={12, 32, 5}
8	$\overline{a}$ ={4, 8, 1}	$\overline{b}$ ={3, 5, 7}
9	$\overline{a}$ ={13, 21, 8}	$\overline{b}$ ={8, 34, 53}
10	$\overline{a}$ ={5, 12, 23}	$\overline{b}$ ={1, 12,44}

Задание 3

Вычислить: $A+B$; $\alpha(B+A)$; $A\cdot A^{-1}$; det A; A^T

Вариант	A	B	α
1	$\begin{matrix} 2 & 4 & 5 \\ 12 & 9 & 3 \\ 1 & 11 & 2 \end{matrix}$	$\begin{matrix} 3 & 0 & 12 \\ 13 & 2 & 7 \\ 1 & 3 & 11 \end{matrix}$	4
2	$\begin{matrix} 6 & 11 & 3 \\ 1 & 9 & 2 \\ 1 & 4 & 7 \end{matrix}$	$\begin{matrix} 2 & 4 & 5 \\ 12 & 9 & 3 \\ 1 & 11 & 2 \end{matrix}$	2
3	$\begin{matrix} 1 & 12 & 7 \\ 3 & 5 & 1 \\ 4 & 6 & 10 \end{matrix}$	$\begin{matrix} 12 & 3 & 7 \\ 4 & 5 & 22 \\ 3 & 8 & 11 \end{matrix}$	6
4	$\begin{matrix} 3 & 24 & 5 \\ 23 & 1 & 0 \\ 3 & 5 & 1 \end{matrix}$	$\begin{matrix} 2 & 4 & 5 \\ 12 & 9 & 3 \\ 1 & 11 & 2 \end{matrix}$	1

Вариант	A	B	α
5	$\begin{matrix} 4 & 3 & 21 \\ 5 & 11 & 13 \\ 1 & 3 & 5 \end{matrix}$	$\begin{matrix} 8 & 1 & 3 \\ 5 & 9 & 19 \\ 10 & 12 & 33 \end{matrix}$	4
6	$\begin{matrix} 11 & 3 & 5 \\ 9 & 2 & 12 \\ 1 & 4 & 23 \end{matrix}$	$\begin{matrix} 32 & 8 & 7 \\ 1 & 11 & 9 \\ 4 & 3 & 5 \end{matrix}$	12
7	$\begin{matrix} 2 & 4 & 5 \\ 12 & 9 & 3 \\ 1 & 11 & 2 \end{matrix}$	$\begin{matrix} 1 & 12 & 7 \\ 3 & 5 & 1 \\ 4 & 6 & 10 \end{matrix}$	10
8	$\begin{matrix} 4 & 3 & 21 \\ 5 & 11 & 13 \\ 1 & 3 & 5 \end{matrix}$	$\begin{matrix} 3 & 0 & 12 \\ 13 & 2 & 7 \\ 1 & 3 & 11 \end{matrix}$	16
9	$\begin{matrix} 2 & 4 & 5 \\ 12 & 9 & 3 \\ 1 & 11 & 2 \end{matrix}$	$\begin{matrix} 11 & 3 & 5 \\ 9 & 2 & 12 \\ 1 & 4 & 23 \end{matrix}$	3
10	$\begin{matrix} 8 & 1 & 3 \\ 5 & 9 & 19 \\ 10 & 12 & 33 \end{matrix}$	$\begin{matrix} 4 & 3 & 21 \\ 5 & 11 & 13 \\ 1 & 3 & 5 \end{matrix}$	11

Задание 4

Отсортировать массив по возрастанию, убыванию, найти минимальное и максимальное значение

Вариант	*массив*
1	1, 15, 23, 25, 123, 18, 21, 40, 22, 54
2	88, 32, 45, 321, 54, 12, 78, 45, 45, 11
3	51, 56, 67, 57, 36, 79, 123, 654, 45, 12
4	43, 76, 98, 65, 43, 96, 92, 100, 890, 123
5	76, 17, 48, 19, 17, 14, 54, 76, 34, 87
6	78, 45, 11, 56, 67, 57, 36, 7, 12, 1
7	67, 43, 76, 32, 97, 31, 74, 89, 32, 43
8	3, 54, 76, 86, 31, 65, 87, 98, 324, 45
9	6, 43, 98, 13, 54, 43, 76, 32, 312, 123
10	56, 43, 76, 43, 87, 12, 41, 90, 103, 43

Задание 5

Построить график функции в полярных координатах

Вариант	*массив*
1	$r = \sin(3\varphi) \quad 0 \le \varphi \le 2\pi$
2	$r = \sin(2\varphi) \quad 0 \le \varphi \le 2\pi$
3	$r = \sin(4\varphi) \quad 0 \le \varphi \le 2\pi$
4	$r = \sin(8\varphi) \quad 0 \le \varphi \le 2\pi$
5	$r = \cos(2\varphi) \quad 0 \le \varphi \le 2\pi$
6	$r = \cos(4\varphi) \quad 0 \le \varphi \le 2\pi$
7	$r = \cos(8\varphi) \quad 0 \le \varphi \le 2\pi$
8	$r = 1 + \cos(\varphi) \ -\pi \le \varphi \le \pi$
9	$r = 1 + \sin(\varphi) \ -\pi \le \varphi \le \pi$
10	$r = \cos(3\varphi) \quad 0 \le \varphi \le 2\pi$

Задание 6

Найти корни многочлена и построить график, проверить полученный результат

1. $P(x) = 3x^4 - 4x^3 - 49x^2 + 64x + 16$
2. $P(x) = 4x^5 - 5x^4 - 11x^3 + 23x^2 - 13x + 2$
3. $P(x) = x^5 - 6x^4 - 3x^3 + 88x^2 - 204x + 144$
4. $P(x) = 2x^5 - 9x^4 + 8x^3 + 15x^2 - 28x + 12$
5. $P(x) = 3x^4 + 5x^3 - 9x^2 - 9x + 10$
6. $P(x) = 2x^4 - 3x^3 - x^2 + 3x - 1$
7. $P(x) = 2x^5 - 9x^4 + 8x^3 + 15x^2 - 28x + 12$
8. $P(x) = 3x^4 + 5x^3 - 9x^2 - 9x + 10$
9. $P(x) = 5x^3 - 12x^2 + 47x - 60$
10. $P(x) = 2x^4 - 3x^3 - x^2 + 3x - 1$