

Национальный исследовательский университет «МИЭТ»

**Лабораторная работа №1.
Основы работы в среде Matlab.**

Москва 2015

Цель работы:

1. Изучение основных приемов работы в среде MatLAB.
2. Получение навыков работы с матрицами и векторами.
3. Изучение способов построения графиков (2-мерных и 3-х мерных).
4. Получение практических навыков использования m-файлов.

Теоретическая часть.

MATLAB – это система математических вычислений, построенная на численных методах (для символьных вычислений используется ядро), а также высокоуровневый язык программирования и среда разработки, позволяющий писать высокоэффективные программы для решения математических задач, в том числе с использованием графического интерфейса. В целом MATLAB — это уникальная коллекция реализаций современных численных методов компьютерной математики, созданных за последние три десятка лет. Она вобрала в себя и опыт, правила и методы математических вычислений, накопленные за тысячи лет развития математики. Возможности системы весьма обширны. Этому способствует не только расширенный набор матричных и иных операций и функций, но и наличие пакета расширения, специально предназначенного для решения задач блочного моделирования динамических систем и устройств, а также десятков других пакетов расширений. В обширном и постоянно пополняемом комплексе команд, функций и прикладных программ (пакетов расширения, пакетов инструментов) системы MATLAB содержатся специальные средства для электротехнических и радиотехнических расчетов (операции с комплексными числами, матрицами, векторами и полиномами, обработка данных, анализ сигналов и цифровая фильтрация), обработки изображений, реализации нейронных сетей, а также средства, относящиеся к другим новым направлениям науки и техники (например, нейронные сети).

Интерфейс системы

Предполагается, что MATLAB используется в операционной системе Windows. После запуска появляется основное окно системы MATLAB (рис. 1). Это обычное окно приложений его можно перемещать, изменять в размерах, открывать на весь экран и т. д. В окне MATLAB размещены командное окно

(Command window), браузер рабочей области (Workspace), браузер файловой системы (Path Browser), который запускается при обычной загрузке системы. Для запуска браузера файловой системы используется окно Current Directory (Текущая папка).

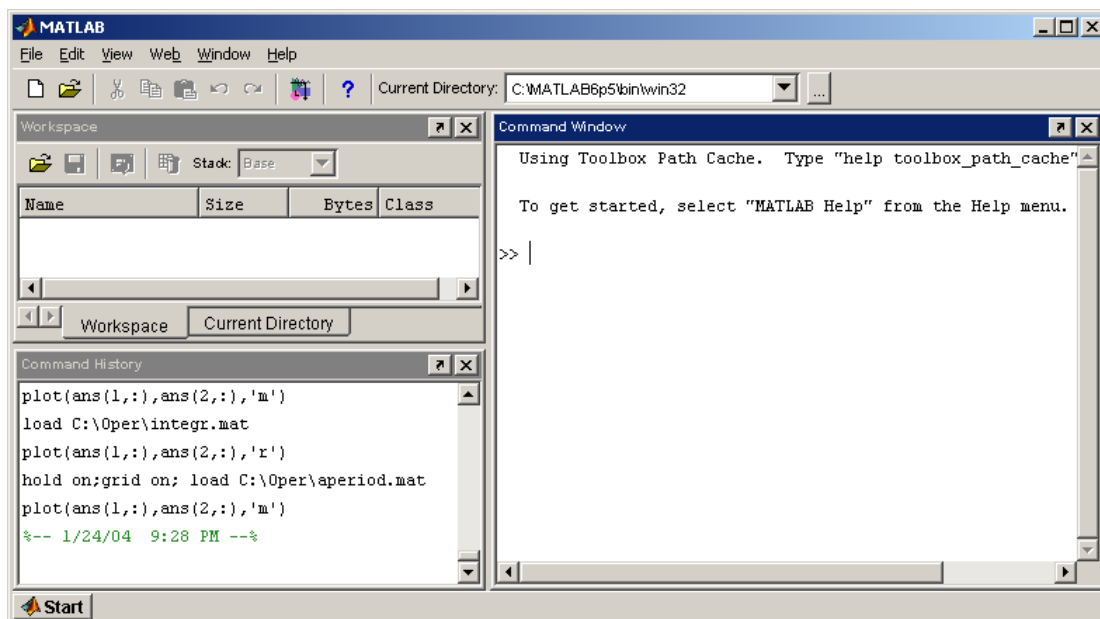


Рис. 1 Окно системы MATLAB после запуска

Основное меню системы MATLAB 6.0. Это меню Рис. 1 сверху шесть пунктов:

- File — работа с файлами;
- Edit — редактирование сессии;
- View — вывод и скрытие панели инструментов;
- Web — доступ к Интернет-ресурсам;
- Windows — установка Windows-свойств окна;
- Help — доступ к справочным подсистемам.

Панель инструментов дает наиболее простой и удобный (особенно для начинающих пользователей) способ работы с системой MATLAB. Кнопки имеют изображение, явно подсказывающее их назначение.

Командное окно (Command window) предназначено для ввода команд. Как отмечалось, в левой части окна системы MATLAB имеется окно специального браузера рабочей области — Workspace Browser. Он служит для просмотра ресурсов рабочей области памяти. Браузер дает наглядную визуализацию содержимого рабочей области. Окно браузера рабочей области выполняет и другие важные функции — позволяет просматривать существующие в памяти объекты, редактировать их содержимое и удалять объекты из памяти. Для вывода содержимого объекта достаточно выделить его имя с помощью мыши и щелкнуть

на кнопке Open (Открыть). Объект можно открыть и двойным щелчком на его имени в списке.

Для просмотра файловой структуры MATLAB служит **браузер файловой системы** (Path Browser). Браузер просмотра файловой структуры позволяет детально ознакомиться с файловой системой MATLAB и вывести любой из m-файлов или файлов демонстрационных примеров для просмотра, редактирования и модификации.

Сеанс работы с MATLAB принято именовать сессией (session). Сессия, в сущности, является текущим документом, отражающим работу пользователя с системой MATLAB. В ней имеются строки ввода, вывода и сообщений об ошибках. Входящие в сессию определения переменных и функций, расположенные в рабочей области памяти, но не саму сессию, можно записать на диск (файлы формата *.mat*), используя команду **save** (Сохранить). Команда **load** (Загрузить) позволяет считать с диска данные рабочей области. Фрагменты сессии можно оформить в виде дневника с помощью команды *diary* (Дневник). При работе с MATLAB в командном режиме действует простейший строчный редактор. Обратите особое внимание на применение клавиш вверх и вниз. Они используются для подстановки после маркера строки ввода >> ранее введенных строк, например для их исправления, дублирования или дополнения. При этом указанные клавиши обеспечивают перелистывание ранее введенных строк снизу вверх или сверху вниз. Такая возможность существует благодаря организации специального стека, хранящего строки с исполненными ранее командами.

Полезно сразу усвоить некоторые команды управления окном командного режима:

clc — очищает экран и размещает курсор в левом верхнем углу пустого экрана.

home — возвращает курсор в левый верхний угол окна.

echo — меняет режим вывода на противоположный.

echo on all — включает режим вывода на экран текста всех m-файлов.

echo off all — отключает режим вывода на экран текста всех m-файлов.

more on — включает режим постраничного вывода (полезен при просмотре больших m-файлов).

more off — отключает режим постраничного вывода (в этом случае для просмотра больших файлов надо пользоваться линейкой прокрутки).

Первые навыки работы.

Система MATLAB создана таким образом, что любые (подчас весьма сложные) вычисления можно выполнять в режиме прямых вычислений, то есть без подготовки программы. Это превращает MATLAB в необычайно мощный калькулятор, который способен производить не только обычные для калькуляторов вычисления (например, выполнять арифметические операции и вычислять элементарные функции), но и операции с векторами и матрицами, комплексными числами, рядами и полиномами. Можно почти мгновенно задать и вывести графики различных функций — от простой синусоиды до сложной трехмерной фигуры.

Работа с системой в режиме прямых вычислений носит диалоговый характер и происходит по правилу «задал вопрос, получил ответ». Пользователь набирает на клавиатуре вычисляемое выражение, редактирует его (если нужно) в командной строке и завершает ввод нажатием клавиши ENTER

Приведем простейший пример: вычисление $2+3$. Просто введите с клавиатуры

```
>>2 + 3
```

и нажмите Enter. Будет выведена надпись: `ans = 5`. MATLAB по умолчанию создаёт переменную с именем `ans`, в которую записывает значение предыдущей операции.

Второй пример. Введите

```
>>a = sin(pi/2)
```

Будет выведено `a = 1`. Создаётся переменная `a`, в которую присваивается значение выражения `sin(pi/2)`. `pi` – предопределённая в MATLAB-е константа π (существует множество других предопределённых констант, например `e`, `i` – мнимая единица).

Если после выражения поставить точку с запятой,

```
>>a = sin(pi/2);
```

то результат вычисления не будет выведен на экран, но переменная будет создана. Её значение можно узнать, дважды щёлкнув по ней в рабочей области (Workspace), либо просто набрав её имя в командной строке:

```
>> a
```

```
a =
```

```
1
```

Полученную переменную можно использовать для дальнейших вычислений, например:

```
>> x = 3 - a  
x =  
2
```

Вы можете использовать стандартные математические операторы (+, -, *, /, ^) и функции, такие как sin, cos, tan (тангенс), cot (котангенс), log (натуральный логарифм), log10(десятичный), log2, exp ... Для изменения порядка выполнения операций используйте (). Аргументы функции задаются в круглых скобках. Записав начальные буквы имени функции, и, нажав клавишу TAB, будет выведен список функций начинающихся с этих букв (если таковые есть) либо слово будет дописано до конца.

```
>> sin  
sin    sinc    single    sinh    sinsml    sinusoid  
sin_tr    sinfo    singvals    sinint    sint    sinv
```

Во время работы вы можете получить справку по интересующей вас функции, набрав

```
>>help имя_функции
```

Работа с векторами и матрицами.

MATLAB это система, работающая с матрицами. Все численные вычисления производятся в матричной форме. Система MATLAB выполняет сложные и трудоемкие операции над векторами и матрицами даже в режиме прямых вычислений без какого-либо программирования. Ею можно пользоваться как мощнейшим калькулятором, в котором наряду с обычными арифметическими и алгебраическими действиями могут использоваться такие сложные операции, как инвертирование матрицы, вычисление ее собственных значений и принадлежащих им векторов, решение систем линейных уравнений, вывод графиков двумерных и трехмерных функций и многое другое. Интересно отметить, что даже обычные числа и переменные в MATLAB рассматриваются как матрицы размера 1x1, что дает единообразные формы и методы проведения операций над обычными числами и массивами. Это также означает, что большинство функций может работать с аргументами в виде векторов и матриц. При необходимости вектора и матрицы преобразуются в массивы, и значения вычисляются для каждого их элемента.

Две записи для вектора — $V=[1\ 2\ 3\ 4]$ и $V=[1,2,3,4]$ — являются идентичными. Таким образом, векторы задаются списком своих элементов, разделяемых пробелами или запятыми. Список заключается в квадратные скобки. Для выделения n -го элемента вектора V используется выражение $V(n)$. Оно задает соответствующую индексированную переменную.

Работа с векторами.

Далее (внимание!) вычисляются функции синуса и экспоненты с аргументом в виде вектора, а не скаляра. В большинстве математических систем вычисление $\sin(x)$ и $\exp(x)$, где x — вектор, сопровождалось бы выдачей ошибки, поскольку функции \sin и \exp должны иметь аргумент в виде скалярной величины. Однако MATLAB — матричная система, а вектор является разновидностью матрицы с размером $1 \times n$. Результат вычислений будет вектором того же размера, что и аргумент x , но элементы возвращаемого вектора будут синусами или экспонентами от элементов вектора x .

```
>> x = 10: -2: 0
```

```
x =
```

```
10 8 6 4 2 0
```

```
>> sin(x)
```

```
ans =
```

```
-0.5440 0.9894 -0.2794 -0.7568 0.9093 0
```

Внимание! При умножении векторов друг на друга получается скаляр:

```
>> y=[2 3 5 8 3 9];
```

$>> x*y$ — здесь допущена ошибка т. к. матрицы (векторы) не согласованны (для умножения матриц $A*B$ должно быть соблюдено правило: число столбцов в матрице A должно равняться числу строк в матрице B):

```
??? Error using ==> *
```

- система выдала сообщение об ошибке

```
Inner matrix dimensions must agree.
```

Правильно будет:

```
>> x*y'
```

```
ans =
```

```
112
```

или можно было создать y как матрицу-столбец: $>> y=[2;3;5;8;3;9]$.

Для умножения (деления, вычитания и т. п.) соответствующих элементов векторов перед оператором (*, / , ^ , + и т. п.) необходимо применять оператор "." (точка). Этот оператор означает поэлементную работу с матрицами.

```
>> z=x.*y
```

```
z =
```

```
20 24 30 32 6 0
```

элементы вектора x умножаются на соответствующие элементы вектора y, обратите внимание на ".".

```
>> z=x.^2
```

 - возведение в квадрат элементов вектора

```
z =
```

```
100 64 36 16 4 0
```

Работа с матрицами.

Для записи матрицы необходимо записать строки, разделенные символом ";", заключенными в квадратные скобки: A=[v1;v2;v3], где v1, v2, v3 –векторы одинаковой размерности. Для работы с матрицами существует множество команд, например:

zeros – создает матрицу, элементами которой являются нули.

ones – создает матрицу, элементами которой являются единицы.

rand – генерирует случайное число, с равномерным законом распределения.

randn – генерирует случайное число, с нормальным законом распределения.

eye(m,n) или **eye([m n])** — возвращают матрицу размера (m x n) с единицами по диагонали и нулями в остальных ячейках.

inv(M) – инвертирование матрицы M (напомним, что обратная матрица определена только для квадратной).

M' – транспонирование M.

det(M) – детерминант матрицы M.

eig(M) – собственные векторы матрицы M (только для квадратной).

Пример:

```
>> Z=zeros(3,5)
```

```
Z =
```

```
0 0 0 0 0
```



```
0 0 0 0 0
0 0 0 0 0
```

Для создания матрицы запишите:

```
>> A=[1 2 3 4; 5 6 7 8; 9 10 11 12]           %матрица A(4x3)
```

или

```
>> A=[1,2,3,4; 5,6,7,8; 9,10,11,12]           %та же самая матрица
```

A =

```
1  2  3  4
5  6  7  8
9 10 11 12
```

Причем элементами матрицы могут быть матрицы или векторы:

```
>> K=[A;1 2 3 4]
```

K =

```
1  2  3  4
5  6  7  8
9 10 11 12
1  2  3  4
```

Вектор является матрицей с числом строк или столбцов равном 1:

```
>> x=[1 2 3 4]
```

x =

```
1  2  3  4
```

Число – матрица из одного элемента:

```
>> c=1.256
```

c =

```
1.2560
```

Над матрицами и элементами матрицы можно выполнять различные операции, но при этом необходимо учитывать правила работы с матрицами.

```
>> B=[2 5; 6 4; 6 5; 8 3]
```

B =

```
2  5
6  4
6  5
8  3
```

Умножение матриц:

```
>> C=A*B           %матрица A задана выше
```

```
C =  
    64    40  
    152   108  
    240   176
```

Умножение соответствующих элементов матриц (необходимо использовать оператор поэлементного доступа "."):

```
>> A1=[1 3 4 6; 9 7 4 0; 8 6 3 9];  
>> C2=A.*A1
```

```
C2 =  
     1     6    12    24  
    45    42    28     0  
    72    60    33   108
```

Добавление элементов ("увеличение матрицы"):

```
>> a1=[A;1 2 3 4] %пара квадратных скобок – оператор объединения.
```

```
a1 =  
     1     2     3     4  
     5     6     7     8  
     9    10    11    12  
     1     2     3     4
```

Транспонирование матрицы (апостроф - '):

```
>> Bt=B'  
Bt =  
     2     6     6     8  
     5     4     5     3
```

Удаление строки (столбца матрицы), в примере удален второй столбец:

```
>> A(:,2)=[]      круглые скобки – оператор извлечения
```

```
A =  
     1     3     4  
     5     7     8  
     9    11    12
```

Извлечение элемента матрицы:

```
>> y=A(3,3) – обращение к элементу матрицы в 3-й строке и 3-м столбце
```

```
y =  
    12
```

или

>> A(5) - обращение по номеру так, если записать все столбцы матрицы друг за другом>

ans =

6

Для извлечения строк или столбцов вместо номера элементов в строке или столбце используется двоеточие, которое является оператором перечисления (с первого до последнего элемента строки или столбца)

>> y=A(3,:) - извлечение 3-й строки из A, двоеточие означает весь набор чисел,

y =

9 11 12

Двоеточие, поставленное между двумя числами – это перечисление от одного до другого числа, с указанным интервалом (по умолчанию 1):

>> x=0:5 %вектор x = (0 1 2 3 4 5)

x =

0 1 2 3 4 5

>> x=0:0.1:0.5 %вектор x = (0 0,2 0,3 0,4 0,5)

x =

0 0.1000 0.2000 0.3000 0.4000 0.5000

Так применение этого оператора при извлечении элементов приводит к извлечению заданных элементов строки или столбца(P(3,2:4) – извлечение со второго по четвертый элементы третьей строки матрицы P).

Работа с графиками.

Одно из достоинств системы MATLAB — обилие средств графики, начиная от команд построения простых графиков функций одной переменной в декартовой системе координат и кончая комбинированными и презентационными графиками с элементами анимации, а также средствами проектирования графического пользовательского интерфейса (GUI). Особое внимание в системе уделено трехмерной графике с функциональной окраской отображаемых фигур и имитацией различных световых эффектов.

Рассмотрим наиболее простые методы построения графиков. Для построения графиков функций "по точкам" (кусочно-линейный график) необходимо

создать вектор из значений независимой переменной с нужным шагом. Например, построение $\sin(x)$:

```
>> x=pi*[-10:0.1:10]; - создание набора переменной x
```

```
>> y=sin(x);
```

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

- построение функцией `plot` графика $y(x)$, `grid` –это построение сетки.

или:

```
>>plot(x,sin(x));grid;
```

Результат выполнения этой команды представлен на рис. 2.

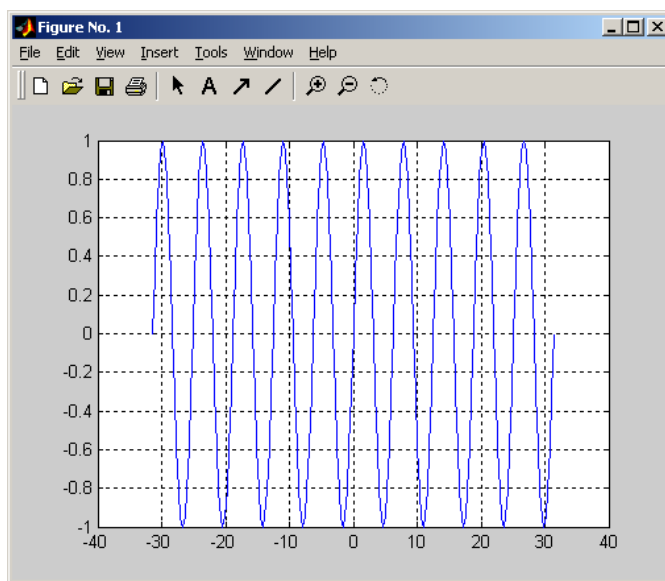


Рис. 2. Пример построения графика синусоиды

для построения в полярной системе координат используйте функцию `polar`:

```
>> polar(x,y);grid;
```

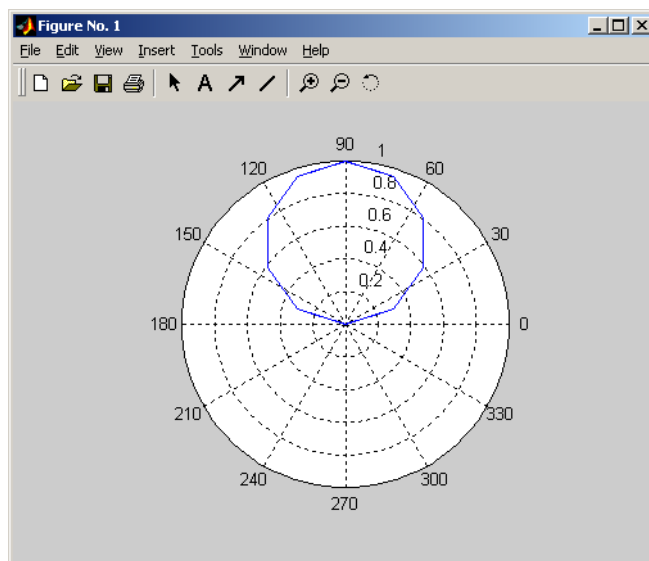


Рис. 3. Результат выполнения команды `polar`

Кроме функций, работающих по "точкам" есть функции, строящие графики без создания векторов переменной (символьно). Они имеют те же имена, но с приставкой "ez" – ezplot, ezpolar и т. п.

```
>> ezplot('sin(x)',[-pi,pi]);grid;
```

- для построения в символьном виде необходимо использовать апострофы, далее идет указание пределов построения.

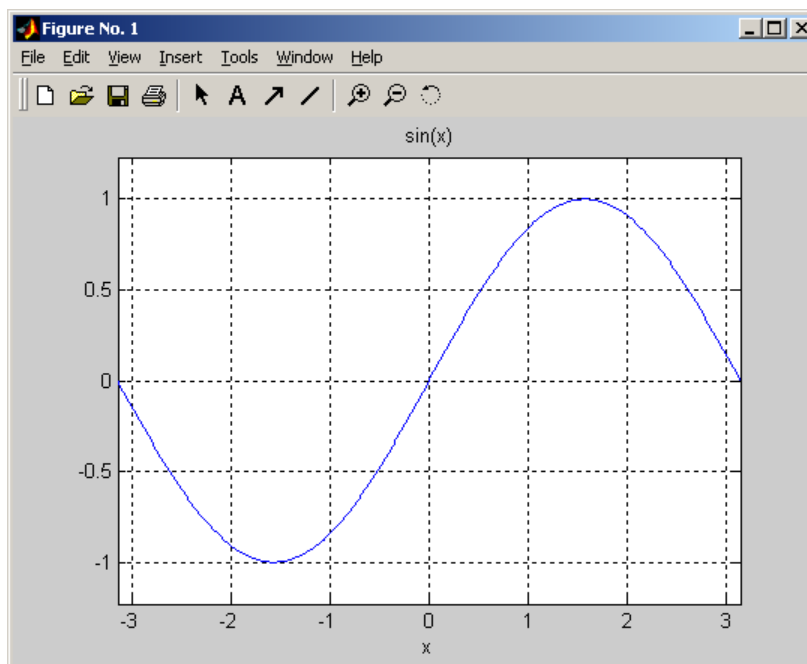


Рис. 4. Результат выполнения команды ezplot

ezplot('x^2+y^2=4') – построит окружность.

Трехмерные поверхности

Трехмерные поверхности обычно описываются функцией двух переменных $z(x, y)$. Специфика построения трехмерных графиков ("по точкам") требует не просто задания ряда значений x и y , то есть векторов x и y . Она требует определения для X и Y двумерных массивов — матриц, которые используются для дальнейших вычислений и построений. Для создания таких массивов служит функция meshgrid. В основном она используется совместно с функциями построения графиков трехмерных поверхностей. Функция meshgrid записывается в следующих формах:

$[X,Y] = \text{meshgrid}(x)$ — аналогична $[X,Y] = \text{meshgrid}(x,x)$;

$[X,Y,Z] = \text{meshgrid}(x,y,z)$ — возвращает трехмерные массивы, используемые для вычисления функций трех переменных и построения трехмерных графиков;

$[X,Y] = \text{meshgrid}(x,y)$ — преобразует область, заданную векторами x и y , в массивы X и Y , которые могут быть использованы для вычисления функции двух

переменных и построения трехмерных графиков. Строки выходного массива X являются копиями вектора x, а столбцы Y — копиями вектора y.

Пример:

```
>> [X,Y]=meshgrid(0:3,-3:0)
```

X =

```
0  1  2  3
0  1  2  3
0  1  2  3
0  1  2  3
```

Y =

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

таким образом, создали квадратные массивы элементов матриц X, Y. Эти данные используются функциями: mesh(X,Y,Z) – построение сетчатого графика, X,Y,Z – матрицы одинаковой размерности; plot3(X,Y,Z) – построение точек, соединенных отрезками прямых (аналог plot); surf(X,Y,Z) – построение поверхности и др.

Пример - построение: $\frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$

<pre>>> [X, Y] = meshgrid(-8:.5:8); >> R = sqrt(X.^2+Y.^2)+eps; >> Z = sin(R)./R; > mesh(X,Y,Z)</pre>	<p>-создание массива значений, по которым вычисляется значение функции в различных точках и вычисление выражения, обратите внимание на ".", поставленную перед операторами – вычисление производятся над элементами матриц.</p>
--	---

Примечание: eps – минимально возможное в MATLAB число, прибавленное для исключения неопределенности sin(0)/0.

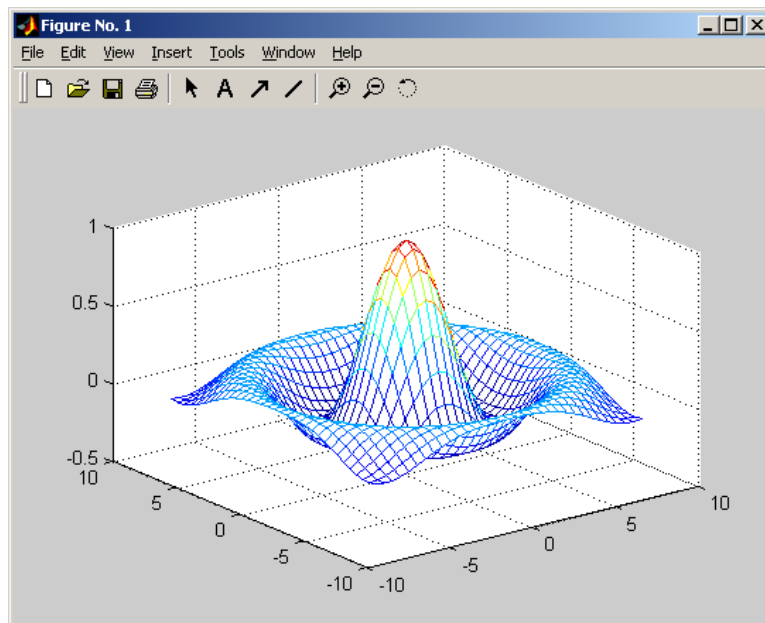


Рис. 5. Результат выполнения команды mesh

Кроме численных методов построения можно использовать символьные: ezmesh, ezsurf, ezplot3 и т. п. Синтаксис использования этих функций прост:

```
>> ezsurf('sin((x^2+y^2)^(0.5))/((x^2+y^2)^(0.5))',[-10 10],[-10 10])
```

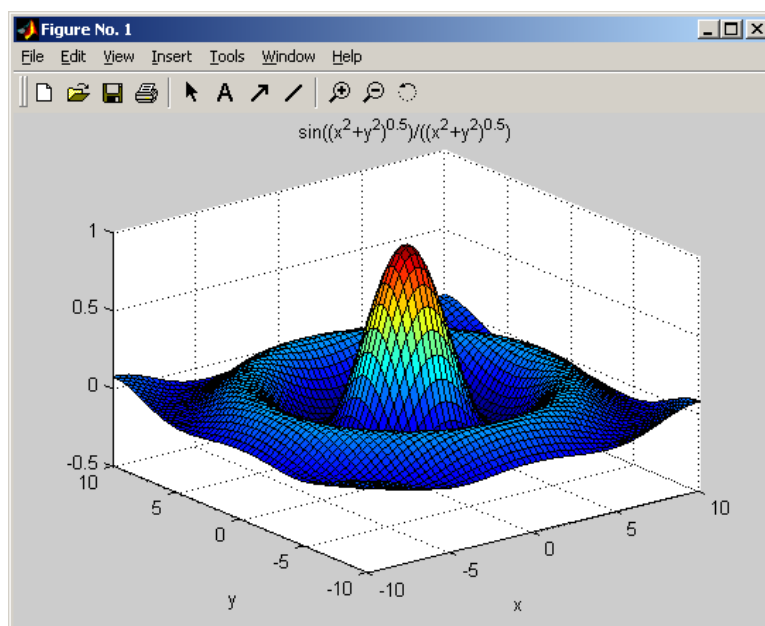


Рис. 6. Результат выполнения команды ezsurf

В апострофы заключается функция, график которой необходимо построить, в скобках указаны пределы (их можно и не задавать). Обратите внимание на отсутствие "." и eps – при использовании данной команды, они просто не нужны. Кроме этого эти функции позволяют строить графики, заданные параметрически, задавать различные параметры графиков и другие возможности (изучите help этих команд).

М-файлы и функции, определяемые пользователем.

Возможности программирования в среде Matlab развиты достаточно хорошо. Они позволяют писать законченные приложения с GUI, но в данной работе мы остановимся только на работе с функциями.

Все функции, которые пользователь создает с тем, чтобы в дальнейшем их использовать, редактируются и хранятся в файлах, имеющих расширение *.m. Для того, чтобы начать писать свою функцию необходимо открыть меню File->New->M-file. Открывается окно редактора m-файлов (в качестве последнего можно использовать обычный Notepad):

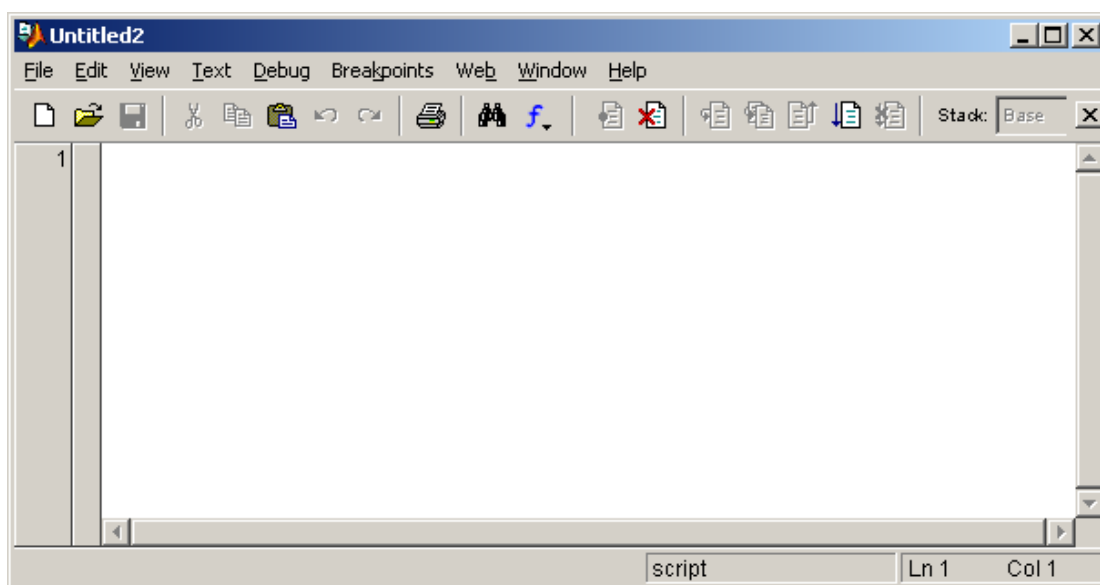


Рис. 7. Внешний вид редактора m-файлов

В этом редакторе пользователь описывает функцию согласно следующим правилам:

Первая строка должна иметь вид: `function y = function_name(x1, x2, ...)`, где `function_name` – имя функции, вызываемой из среды Matlab (в файле могут быть и другие функции, но они носят служебный характер, т.е. из среды Matlab не вызываются).

Замечание: имя m-файла должно быть `function_name.m`. Далее пишется тело функции.

Пример:

Реализовать функцию `my_sin(x,y,z)`, которая будет строить синусоиду с амплитудой (x), фазой (y) и в заданных пределах z.

В редакторе м-файлов описываем требуемую функцию:

```
function f=my_sin(x,y,z)
f=x*sin(z+y);
plot(z,f); grid;
```

Сохраняем файл под именем my_sin.m. В командном окне Matlab пишем

```
my_sin(5,0,[-pi:0.05:pi]);
```

в результате получаем требуемый график

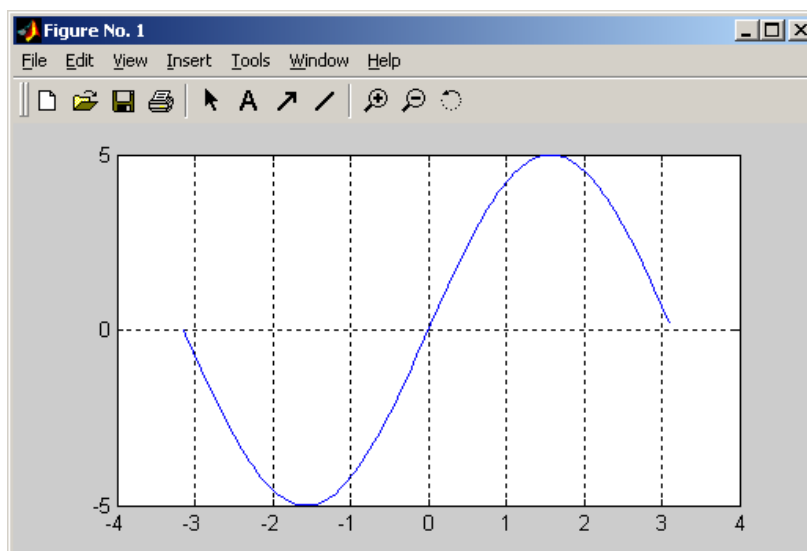


Рис. 8. Результат выполнения функции my_sin(x,y,z)

Важно: для того, чтобы функция выполнялась, необходимо, чтобы рабочая директория совпадала с директорией, в которой сохранен файл my_sin.m.

Практическая часть

Внимание!!! Результаты выполнения упражнений необходимо сохранять в виде ТЕКСТОВОГО файла (*.DOC, *.TXT), который предъявляется преподавателю при защите работы.

Упражнение 1.

Для расширения представления об интерфейсе и возможностях системы выполните следующие задачи:

- в меню View отключите браузер рабочей области и прочие, не нужные сейчас вам окна, для чего выберете View->Desktop layout->Command Window Only.

Упражнение 2:

Вычислите значение выражения $\frac{\ln\left(\sin x \frac{\pi}{8}\right) + x^2 + \sqrt{2x-2}}{2^y}$

для 1) $x = 1,001$, $y = -10,7$; 2) $x = 2+3i$, $y = 1-i$.

Упражнение 3:

1. Создайте вектор X , элементы которого – члены арифметической прогрессии с начальным членом 5, шагом -3, состоящий из 4-х элементов.
2. Вычислите арктангенс его элементов.
3. Возведите его элементы в куб, возведите вектор в квадрат (скалярное умножение).

Упражнение 4:

Создайте три матрицы $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$; $B = \begin{pmatrix} 2 & 6 & 8 \\ 6 & 11 & 0.5 \end{pmatrix}$; $C = (13 \ 87 \ 76 \ 45 \ 44)$

1. умножьте матрицу A на B . Что произошло?
2. объедините A и B^T и C ;
3. из получившейся матрицы извлеките матрицу (3x3), находящуюся в левом верхнем углу;
4. удалите из матрицы, полученной в задании 2, четные строки и столбцы;
5. решите систему уравнений методом обратной матрицы:

$$2x+5y-8z=8$$

$$5x+6y+3z=12$$

$$4x-5y-z=23$$

Упражнение 5:

1. постройте графики $y = x - \sin(3x)$; $y = \frac{x}{\sin(x)}$ двумя способами. Для второго графика см. функцию fplot;
2. постройте $y=x*\sin(5x)$ в полярной СК, для x от 0 до 2π и от -2π до 2π ;
3. постройте эллипс с центром координат $(-3,2)$ большей полуосью 5, малой 4, при помощи plot, ezplot;

4. посмотрите в меню tools и edit, что можно сделать с графиком, поменяйте его цвет, назовите оси координат.

Упражнение 6:

1. постройте график функции $y=f(x)$, где $x=\sin(t)$, $y=2*\cos(2t)$, $-5<t<5$;
2. постройте график функции $z = \frac{x^2 + y^2}{5 \times x \times y} + \ln(x + y)$ для $0 < x < 3$; $0 < y < 3$;

Упражнение 7:

1. Самостоятельно изучить операторы циклов for, while, if, switch.

Привести примеры их использования.

2. Написать функцию, которая считает факториал своего параметра, если введено отрицательное число, должно выдаваться сообщение об ошибке.

Подсказка: требуемая функция может выглядеть следующим образом `calc_fact(n)`; также смотри `help disp`.

3. Написать функцию, которая считает сумму ряда: $f_n=1/n^{(N+1)}$, где N – номер варианта, с заданной точностью (точность вводит пользователь).

Подсказка: смотри `help input`.