



EN 161

MATLAB Session



Ibrahim Eden
ieden@lems.brown.edu

MATrix LABoratory

- environment for scientific (mathematical) computation
- workspace: commandline, editor, figures
- platform independent



The Most Useful Function

HELP Command:

Example:

```
>> help imread
```

IMREAD Read image from graphics file.

A = IMREAD(FILENAME,FMT) reads a grayscale or color image from the file specified by the string **FILENAME**, where the string **FMT** specifies the format of the file. See the reference page, or the output of the function **IMFORMATS**, for a list of supported formats. If the file is not in the current directory or in a directory in the **MATLAB** path, specify the full pathname of the location on your system. If **IMREAD** cannot find a file named **FILENAME**, it looks for a file named **FILENAME.FMT**.

.



Useful Commands

| | |
|------------|--|
| % | Used to indicate comment |
| ; | If used at the end of a line, MATLAB does not display the result |
| +, -, *, / | Arithmetic operators |
| .*, ./, .^ | Element wise operators |



Defining Variables

| | |
|---------------------------|------------------------------------|
| <code>a=5;</code> | Used to define a scalar |
| <code>b=[1 2; 3 4]</code> | Define a matrix (2x2) |
| <code>v=[1 2 3 4];</code> | Define a row vector |
| <code>v=[1;2;3;4]</code> | Define a column vector |
| <code>v=1:0.01:2</code> | A vector filled in specified range |
| <code>v=[]</code> | Empty vector |



Creating Special Matrices

| | |
|---------------------------|-----------------------------|
| <code>m=zeros(3)</code> | Creates 3x3 matrix of zeros |
| <code>m=zeros(3,2)</code> | Creates 3x2 matrix of zeros |
| <code>v=ones(1,3)</code> | A row vector of ones |
| <code>v=rand(3,4)</code> | A random 3x4 matrix |



Accessing Matrix Elements

| | |
|---|---------------------------------|
| <code>v(3)</code> | A vector element |
| <code>m(3,4)</code> | A matrix element |
| <code>m(2,:)</code> or <code>m(:,4)</code> | A Matrix row A matrix column |
| <code>m(1:4,2:5)</code> | A matrix region |
| <code>size(m)</code> | Size of the matrix |



Some Useful Functions

| | |
|------|---|
| sum | Sum of elements in vector Sum of columns of a matrix |
| mean | ... |
| var | ... |
| max | ... |
| min | ... |
| inv | Inverse of a matrix |



Control Statements

| | |
|---|---|
| for VARIABLE = EXPR STATEMENT ... STATEMENT end | K=0; for i=1:10 K = K + i; end |
| while EXPR STATEMENT ... STATEMENT end | i=1; K=0; while i<11 K=K+i; i=i+1; End |
| if EXPR STATEMENTS elseif EXPR STATEMENTS else STATEMENTS end | if i=0 a=1; elseif i=1 a=2; else a=3; End |



Creating Functions (.m files)

Functions are also m-files.

The first line in a function file must be of this form:

```
function [outarg_1, ..., outarg_m] = myfunction(inarg_1, ..., inarg_n)
```

myfunction.m

```
function y = myfunction(x)
```

```
% Function of one argument with one return value
```

```
a = [-2 -1 0 1];
```

```
y = a + x;
```

myotherfunction.m

```
function [y, z] = myotherfunction(a, b)
```

```
% Function of two arguments with two return values
```

```
y = a + b;
```

```
z = a - b;
```



Loading/Saving Images

| | |
|---------------------------------|--|
| <code>I=imread('path')</code> | Reads the image in the specified path and loads it into a matrix |
| <code>imwrite(im,'path')</code> | Saves the image in matrix "im" to the specified path |
| <code>imshow(im)</code> | Displays the image in the specified matrix |



Some Hints

>> help ginput

GINPUT Graphical input from mouse.

$[X,Y] = \text{GINPUT}(N)$ gets N points from the current axes and returns the X- and Y-coordinates in length N vectors X and Y . The cursor can be positioned using a mouse (or by using the Arrow Keys on some systems). Data points are entered by pressing a mouse button or any key on the keyboard except carriage return, which terminates the input before N points are entered.



Some Hints

```
>> help ginput
```

GINPUT Graphical input from mouse.

[X,Y] = GINPUT(N) gets N points from the current axes and returns the X- and Y-coordinates in length N vectors X and Y. The cursor can be positioned using a mouse (or by using the Arrow Keys on some systems). Data points are entered by pressing a mouse button or any key on the keyboard except carriage return, which terminates the input before N points are entered.

Double vs. UINT8: imread function loads images in the UINT8 type (8-bit integer which can take values between 0-255).

MATLAB will not allow you to perform arithmetic operations on UINT8 variables so you have to convert your image to double type

```
>> imD = double(im);
```



Some Hints

>> help plot

PLOT Linear plot.

PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up. If X is a scalar and Y is a vector, length(Y) disconnected points are plotted.

PLOT(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:

| | | | | | |
|---|-------|---|--------|----|---------|
| b | blue | . | point | - | solid |
| g | green | o | circle | : | dotted |
| r | red | x | x-mark | -. | dashdot |



Some Hints

>> help hold

HOLD Hold current graph

HOLD ON holds the current plot and all axis properties so that subsequent graphing commands add to the existing graph.

HOLD OFF returns to the default mode whereby PLOT commands erase the previous plots and reset all axis properties before drawing new plots.



Some Hints

```
>> help hist
```

HIST Histogram.

$N = \text{HIST}(Y)$ bins the elements of Y into 10 equally spaced containers and returns the number of elements in each container. If Y is a matrix, HIST works down the columns.

$N = \text{HIST}(Y,M)$, where M is a scalar, uses M bins.

Important update:

If you want to see the plot of histogram, you should use:

```
>> hist(y,m)
```

If you want the number of elements in the bins, use:

```
>> a = hist(y,m)
```

